UNIVERSITI TEKNOLOGI MALAYSIA

SCHOOL OF COMPUTING
Faculty of Engineering

SECR2033 – COMPUTER ORGANIZATION AND ARCHITECTURE

PROJECT REPORT

FINAL

COURSE NAME: BACHELOR OF COMPUTER SCIENCE–COMPUTER NETWORKS & SECURITY [1SECR]

STUDENTS' NAME	METRIC NUMBER	SECTION
ARIF BIN SUHAIMI	A19EC0023	03
MUHAMMAD AMIN QAYYUM BIN MOHD RAZI	A19EC0093	03
MUHAMMAD ISKANDAR ZULQARNAIN BIN MOHD ISHAK	A19EC0098	04

LECTURER'S NAME: MS. MARINA BINTI MD ARSHAD

Table of Contents

Member Responsibilities 2
Introduction 3
Game Design 4
Input and Output Example 6
Code Explanation 10
Discussions 12
Conclusion 13
References 14
Appendix 15

Member Responsibilities

Members	Responsibilities
MUHAMMAD ISKANDAR ZULQARNAIN BIN MOHD ISHAK	-Introduction -Game Design -Input Output -Design animation -Make statements for each fruit
ARIF BIN SUHAIMI	-Coding -Coding explanation -Design layout -Design animation
MUHAMMAD AMIN QAYYUM BIN MOHD RAZI	-Discussion -Conclusion -Design animation -Make statements for each fruit

Introduction

Computer's programming language has been a core of computers software being built by those who are master on that field. We have many languages related to the computer programming like C, C++, Java, Pascal etc. Besides, we also have Assembly Language.

Assembly Language allows the computer programmer to develop a lot of handy system to ease the user and enhance users' experience. Not only constrained by those components, Assembly Language can also be used to create simple games using its syntax and libraries' component to perform certain animations. Therefore, to fulfil our course requirement in Computer Organization and Architecture, my team and I are required to come out with a simple and captivating game using the Assembly Language. We used Microsoft Visual Studio as our tool since we were exposed to this kind of Integrated Development Environment (IDE) during the course's programming session.

In this report, we define the layouts and ideas of our game from the very first step of brainstorming the flow, ideas, and design of the game until the final game's interface. Every part of this project will be elaborated in detail to give the clear illustration of our creation. Plus, this project also allows us to enhance our skills and talents in writing the codes by embedding a few syntaxes that we rarely found during lab session in class. These syntaxes were allocated for certain functions to be utilized in our game project.

Game Design

By referring to game design theory, we found out that there are three principles to be followed to come out with astounding ideas. The principles include building around a core game mechanic, easy to learn but fun to master, and reward the player. Since that this game designation is for academic purposes, we only consider two of it, first and third values.

Building a core game mechanic is when we figure out the outline and produces the flowchart of how the game is going to be run. Picking up a ‘Fruits’ theme, we then develop the entire game with variety of fruits included. “Perplexing Crate” is our proposed title of the game. Literally, it is a synonym to “Mysterious Box”. So, when player first time try our diversion, they will get the gist of what is the game is all about – finding something behind those enigmatic packages. The essence of the game includes some life aspects such as emotions, future, and people’s opinion. These bases will bring you further into the game and you may find out what is in the crates.

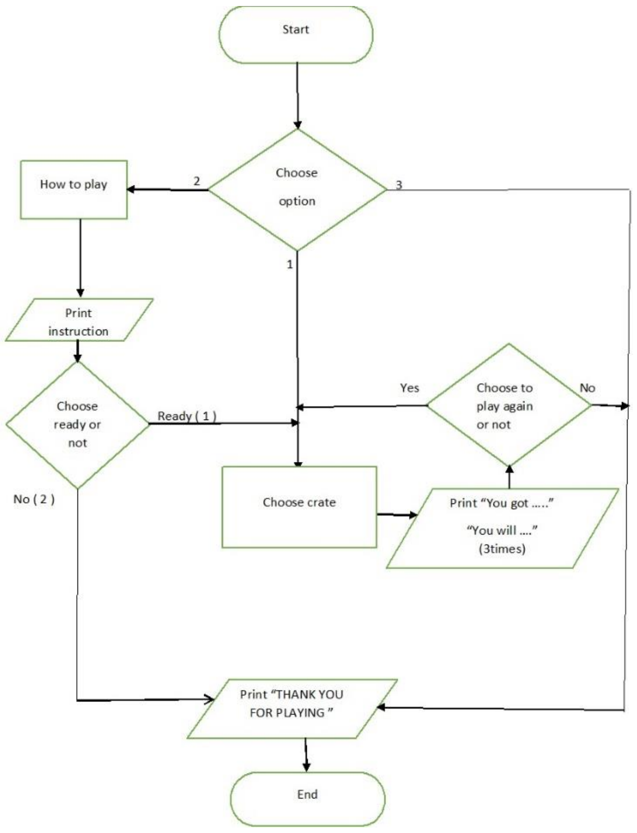
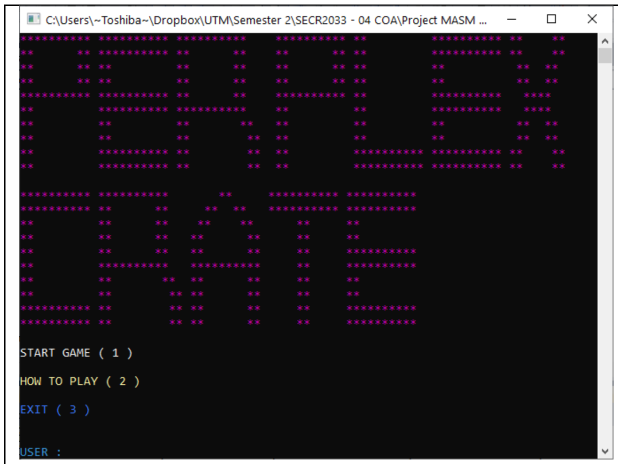


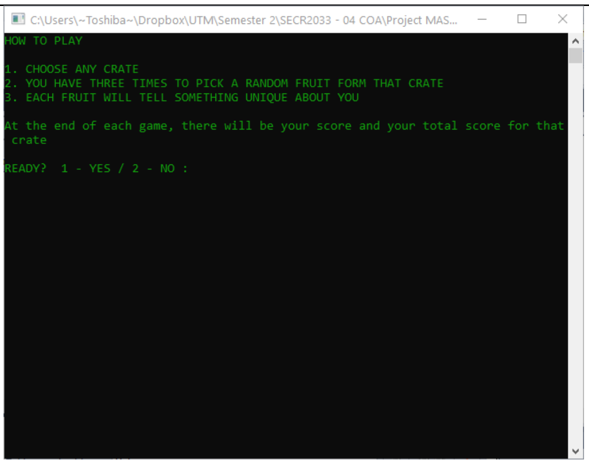
Figure 1 shows our game's flowchart

As the game runs and play, there will be several output screens that will be displayed to the player during the process. Plus, there are also some animations that we embed to enhance user experience while playing Perplexing Crate. The screens are shown in Table 1 below.



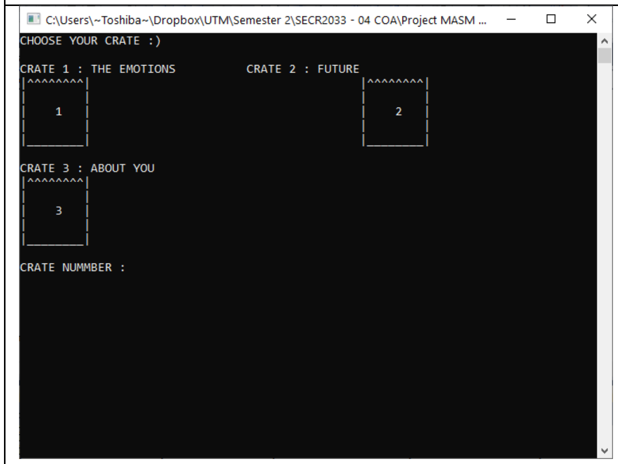
```
C:\Users\~Toshiba~\Dropbox\UTM\Semester 2\SECR2033 - 04 COA\Project MASM ...

START GAME ( 1 )

HOW TO PLAY ( 2 )

EXIT ( 3 )


USER :
```

This is a Welcome screen of our game. It will be showing three options stated there. There is also an animation on the game's title 'Perplexing Crate' as it may change during game's boot up.



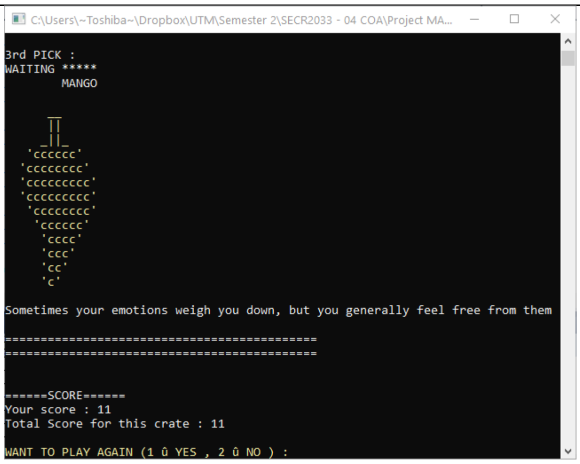
```
C:\Users\~Toshiba~\Dropbox\UTM\Semester 2\SECR2033 - 04 COA\Project MAS...
HOW TO PLAY

1. CHOOSE ANY CRATE
2. YOU HAVE THREE TIMES TO PICK A RANDOM FRUIT FORM THAT CRATE
3. EACH FRUIT WILL TELL SOMETHING UNIQUE ABOUT YOU

At the end of each game, there will be your score and your total score for that
 crate

READY?  1 - YES / 2 - NO :
```

When player choose option (2) HOW TO PLAY then they will be brought to help screen whereby all the instruction of playing the game are explained here.



```
C:\Users\~Toshiba~\Dropbox\UTM\Semester 2\SECR2033 - 04 COA\Project MASM ...
CHOOSE YOUR CRATE :)

CRATE 1 : THE EMOTIONS          CRATE 2 : FUTURE
|^^^^^^^^|                                       |^^^^^^^^|
|        |                                       |        |
|   1    |                                       |   2    |
|        |                                       |        |
|________|                                       |________|

CRATE 3 : ABOUT YOU
|^^^^^^^^|
|        |
|   3    |
|        |
|________|

CRATE NUMMBER :
```

To begin the game, user will choose option (1) START GAME. The game will begin by giving three option of crate to be choosen. One at a time.



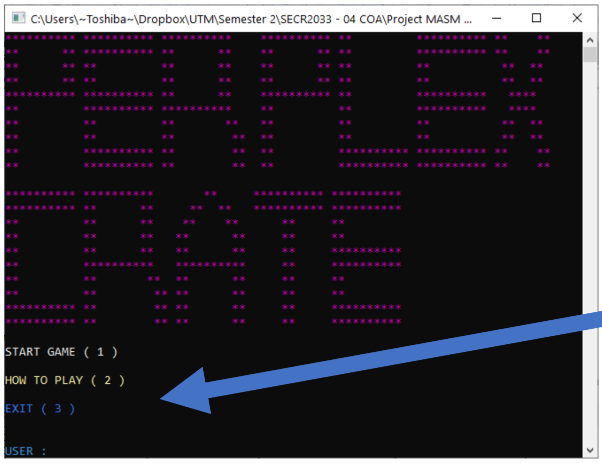
```
C:\Users\~Toshiba~\Dropbox\UTM\Semester 2\SECR2033 - 04 COA\Project MA...

3rd PICK :
WAITING *****
        MANGO


       __
       ||
      _||_
   'cccccc'
  'cccccccc'
  'ccccccccc'
  'ccccccccc'
   'cccccccc'
    'cccccc'
     'cccc'
     'ccc'
     'cc'
     'c'

Sometimes your emotions weigh you down, but you generally feel free from them

============================================
============================================



======SCORE======
Your score : 11
Total Score for this crate : 11

WANT TO PLAY AGAIN (1 û YES , 2 û NO ) :
```

Once the player chosed, they will be redirected to the choosen crate and pick up three fruits by pressing ENTER button on the keyboard. Here the score is also be displayed.

Table 1 shows the output screens of the game

Input and Output Example

In this game, we are making the player to use minimal input action. For example, we only required the player to enter numeric key rather than alphabet and character key. By this way, player may have good experience of playing our diversion with less effort.



```
C:\Users\~Toshiba~\Dropbox\UTM\Semester 2\SECR2033 - 04 COA\Project MASM ...

START GAME ( 1 )

HOW TO PLAY ( 2 )

EXIT ( 3 )

USER :
```

INPUT(S) DATA TYPE: numerical

INPUT(S) VALUE: **1 2 3**

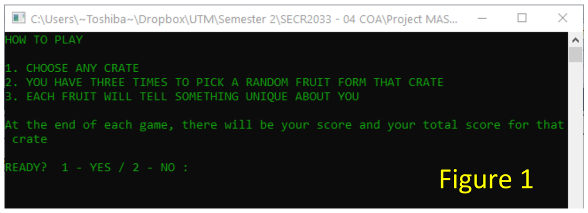
PROCEEDING INPUT: **ENTER**

OUTPUT(S)

IF (INPUT == 1) → Figure 1

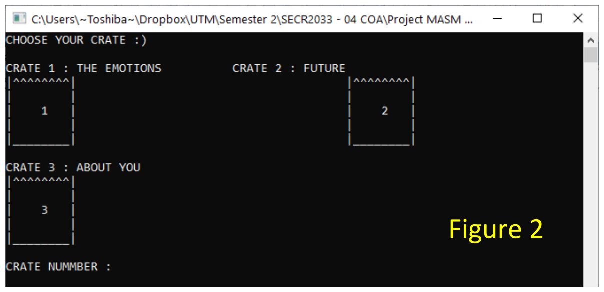
IF (INPUT == 2) → Figure 2

IF (INPUT == 3) → Figure 3



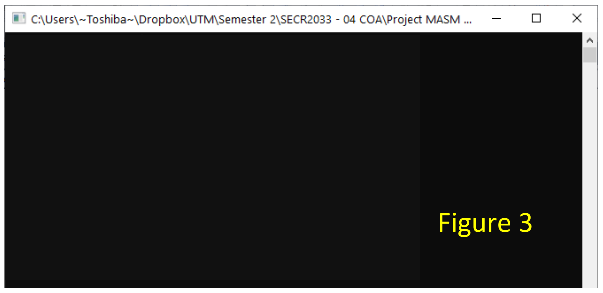
```
C:\Users\~Toshiba~\Dropbox\UTM\Semester 2\SECR2033 - 04 COA\Project MAS...
HOW TO PLAY

1. CHOOSE ANY CRATE
2. YOU HAVE THREE TIMES TO PICK A RANDOM FRUIT FORM THAT CRATE
3. EACH FRUIT WILL TELL SOMETHING UNIQUE ABOUT YOU

At the end of each game, there will be your score and your total score for that
 crate

READY?  1 - YES / 2 - NO :
```

Figure 1



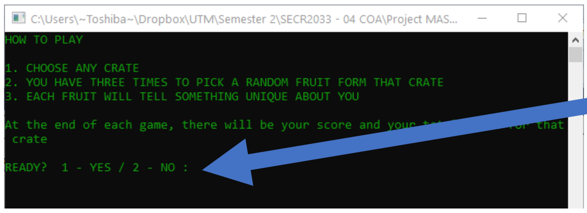
```
C:\Users\~Toshiba~\Dropbox\UTM\Semester 2\SECR2033 - 04 COA\Project MASM ...
CHOOSE YOUR CRATE :)

CRATE 1 : THE EMOTIONS          CRATE 2 : FUTURE
|^^^^^^^^|                                      |^^^^^^^^|
|        |                                      |        |
|    1   |                                      |    2   |
|        |                                      |        |
|________|                                      |________|

CRATE 3 : ABOUT YOU
|^^^^^^^^|
|        |
|    3   |
|        |
|________|

CRATE NUMMBER :
```

Figure 2



```
C:\Users\~Toshiba~\Dropbox\UTM\Semester 2\SECR2033 - 04 COA\Project MASM ...
```

Figure 3



```
C:\Users\~Toshiba~\Dropbox\UTM\Semester 2\SECR2033 - 04 COA\Project MAS...
HOW TO PLAY

1. CHOOSE ANY CRATE
2. YOU HAVE THREE TIMES TO PICK A RANDOM FRUIT FORM THAT CRATE
3. EACH FRUIT WILL TELL SOMETHING UNIQUE ABOUT YOU

At the end of each game, there will be your score and your to          for that
 crate

READY?  1 - YES / 2 - NO :
```

INPUT(S) DATA TYPE: numerical

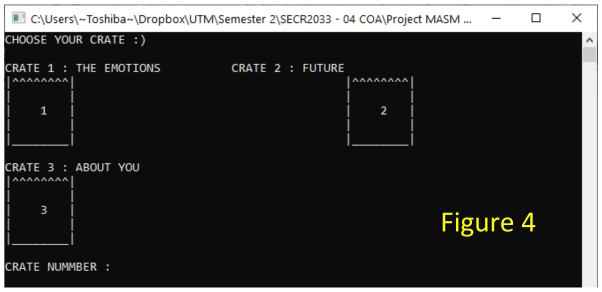
INPUT(S) VALUE: **1 2**

PROCEEDING INPUT: **ENTER**

OUTPUT(S)

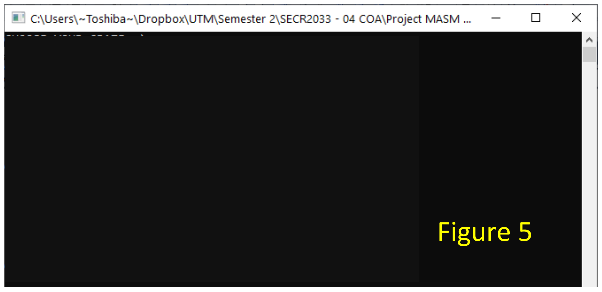
IF (INPUT == 1) → Figure 4

IF (INPUT == 2) → Figure 5



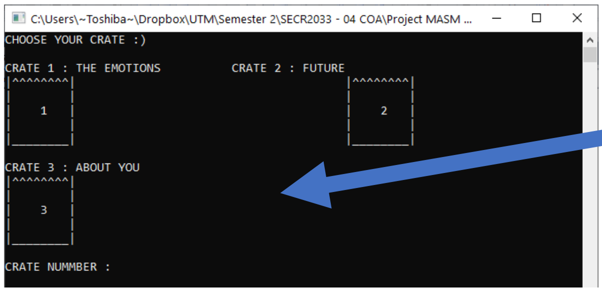
```
C:\Users\~Toshiba~\Dropbox\UTM\Semester 2\SECR2033 - 04 COA\Project MASM ...
CHOOSE YOUR CRATE :)

CRATE 1 : THE EMOTIONS          CRATE 2 : FUTURE
    1                                 2

CRATE 3 : ABOUT YOU
    3

CRATE NUMMBER :
```

Figure 4



```
C:\Users\~Toshiba~\Dropbox\UTM\Semester 2\SECR2033 - 04 COA\Project MASM ...
```

Figure 5



```
C:\Users\~Toshiba~\Dropbox\UTM\Semester 2\SECR2033 - 04 COA\Project MASM ...
CHOOSE YOUR CRATE :)

CRATE 1 : THE EMOTIONS          CRATE 2 : FUTURE
    1                                 2

CRATE 3 : ABOUT YOU
    3

CRATE NUMMBER :
```

INPUT(S) DATA TYPE: numerical

INPUT(S) VALUE: **1 2 3**

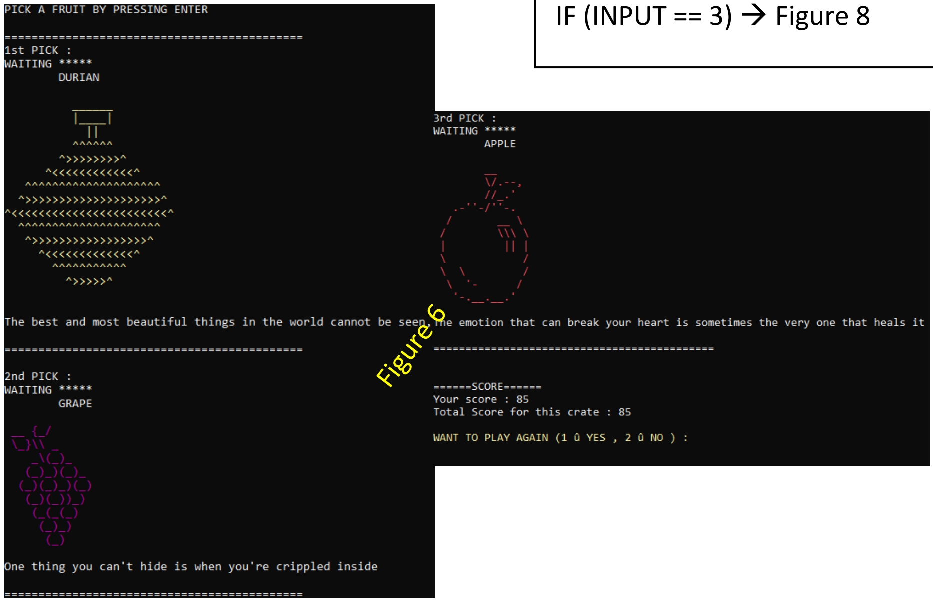
PROCEEDING INPUT: **ENTER**

OUTPUT(S)

IF (INPUT == 1) → Figure 6

IF (INPUT == 2) → Figure 7

IF (INPUT == 3) → Figure 8



```
PICK A FRUIT BY PRESSING ENTER

============================================
1st PICK :
WAITING *****
        DURIAN

The best and most beautiful things in the world cannot be seen

============================================

2nd PICK :
WAITING *****
        GRAPE

One thing you can't hide is when you're crippled inside

============================================
```

```
3rd PICK :
WAITING *****
        APPLE

The emotion that can break your heart is sometimes the very one that heals it

============================================


======SCORE======
Your score : 85
Total Score for this crate : 85

WANT TO PLAY AGAIN (1 û YES , 2 û NO ) :
```

Figure 6

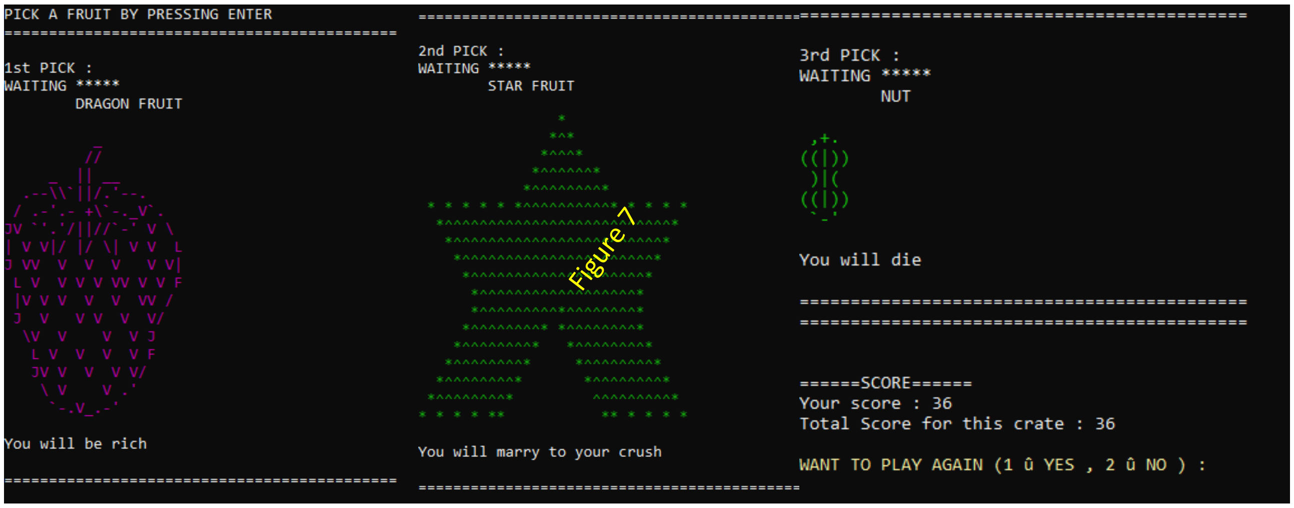


Figure 7

```
PICK A FRUIT BY PRESSING ENTER
=============================================

1st PICK :
WAITING *****
        DRAGON FRUIT

You will be rich

=============================================
=============================================

2nd PICK :
WAITING *****
        STAR FRUIT

You will marry to your crush

=============================================
=============================================

3rd PICK :
WAITING *****
        NUT

You will die

=============================================
=============================================

======SCORE======
Your score : 36
Total Score for this crate : 36

WANT TO PLAY AGAIN (1 û YES , 2 û NO ) :
```

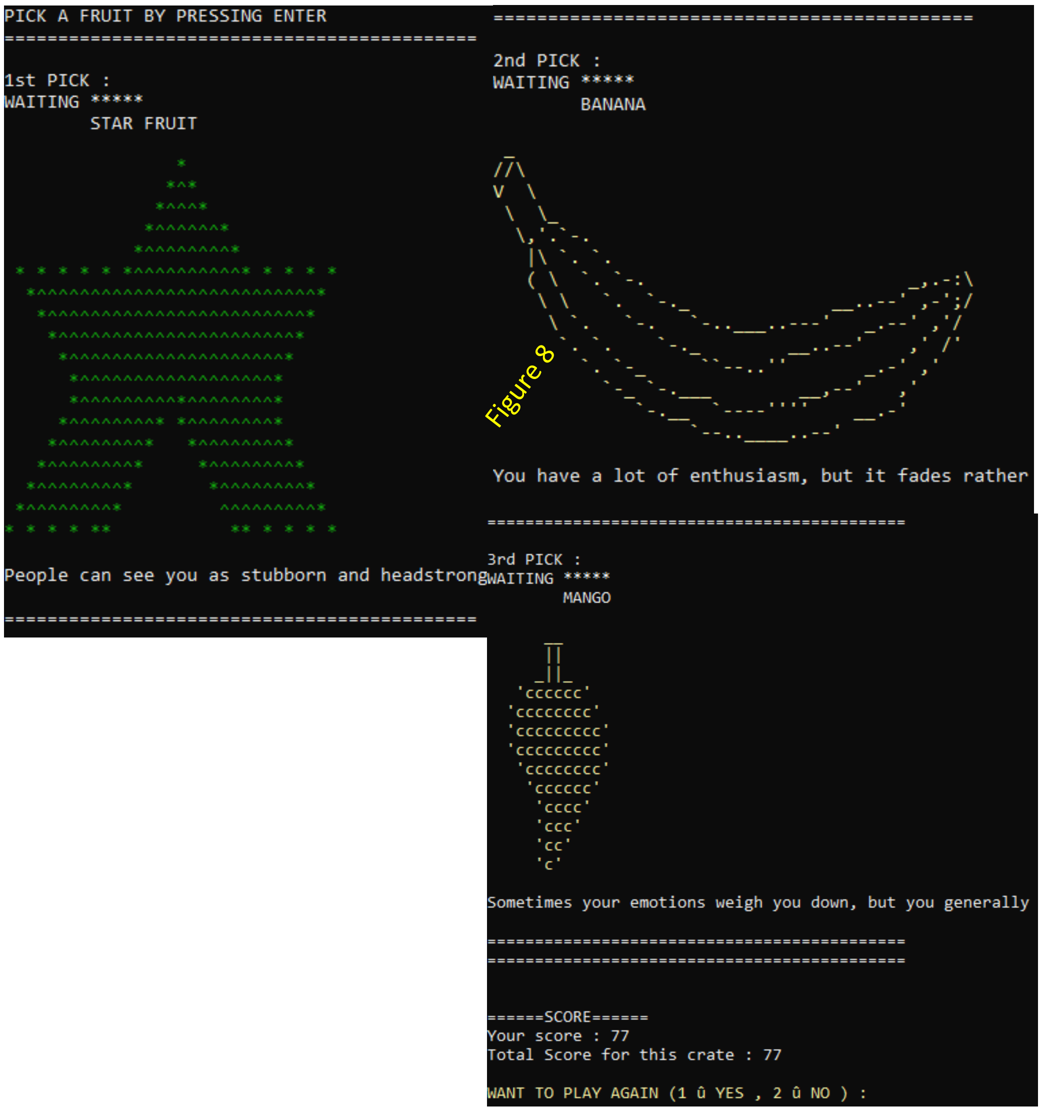
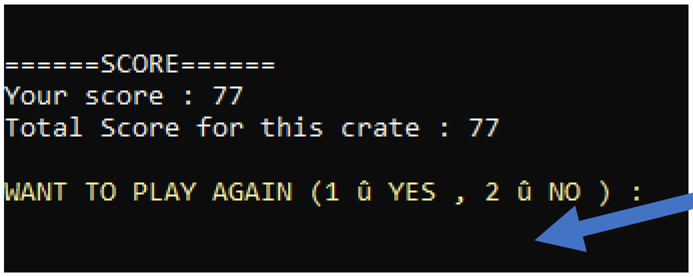


Figure 8

```
PICK A FRUIT BY PRESSING ENTER
=============================================

1st PICK :
WAITING *****
        STAR FRUIT

People can see you as stubborn and headstrong

=============================================
=============================================

2nd PICK :
WAITING *****
        BANANA

You have a lot of enthusiasm, but it fades rather

=============================================

3rd PICK :
WAITING *****
        MANGO

Sometimes your emotions weigh you down, but you generally

=============================================
=============================================

======SCORE======
Your score : 77
Total Score for this crate : 77

WANT TO PLAY AGAIN (1 û YES , 2 û NO ) :
```



```
======SCORE======
Your score : 77
Total Score for this crate : 77

WANT TO PLAY AGAIN (1 û YES , 2 û NO ) :
```

INPUT(S) DATA TYPE: numerical

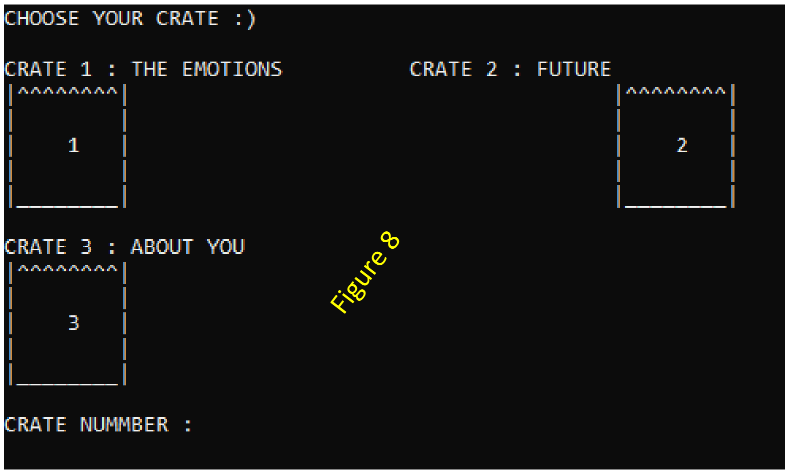
INPUT(S) VALUE: **1 2 3**

PROCEEDING INPUT: **ENTER**

OUTPUT(S)

IF (INPUT == 1) → Figure 8

IF (INPUT == 2) → Figure 9



```
CHOOSE YOUR CRATE :)

CRATE 1 : THE EMOTIONS          CRATE 2 : FUTURE
|^^^^^^^^|                                        |^^^^^^^^|
|        |                                        |        |
|   1    |                                        |   2    |
|        |                                        |        |
|________|                                        |________|

CRATE 3 : ABOUT YOU
|^^^^^^^^|
|        |
|   3    |
|        |
|________|

CRATE NUMMBER :
```

Once the player reaches the part where the game prompted `"WANT TO PLAY AGAIN (1 – YES , 2 – NO ) : "` then the process of input and output begins again following the exact same flow from Welcome screen until the End screen indicating the end of the game.

Code Explanation

1. ret

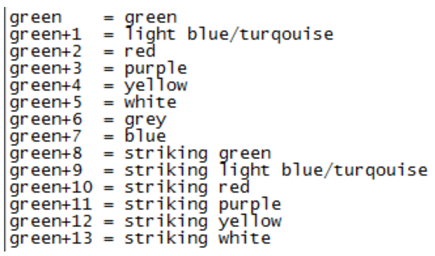
Basically, ret is a return statement like we use in the high-level language. Ret instruction will return to the address when you call the procedure. For example, this code segment:

```
. code
main PROC

       call colour1
       call ani
       call prompt_2
       call prompt_3
       call prompt_4
       call Crlf
       call Crlf
       call prompt_5
       call ReadDec
       cmp eax,1
       je start
       cmp eax,2
       je how
       jmp luar
main ENDP

colour1 proc
       mov ebx,green+7
       mov eax,ebx
       call SetTextColor
       ret
colour1 end
```

"main proc" calls the colour1 procedure. The program will go to "colour1 proc" and execute all the syntax inside the procedure. When it read the ret statement, it will return to "main proc" and then continue read the next syntax after the point where "colour1 proc" called.

2. SetTextColor

The text colours that we use requires a segment of codes. "SetTextColor" works with certain value of EAX register. This is the colour reference for SetTextColor,



```
green    = green
green+1  = light blue/turqouise
green+2  = red
green+3  = purple
green+4  = yellow
green+5  = white
green+6  = grey
green+7  = blue
green+8  = striking green
green+9  = striking light blue/turqouise
green+10 = striking red
green+11 = striking purple
green+12 = striking yellow
green+13 = striking white
```

This is the example:

```
mov ebx,green+7
mov eax,ebx
call SetTextColor
```

The text or anything that display on the terminal will be blue.

3. Delay

Delay statement implement the same function as Sleep() statement in c++. The duration of the delay depends on the value inside EAX (in milliseconds). Example:

```
mov eax,900
call Delay
```

The terminal will delay or in other words stop for 900 milliseconds (0.9 seconds).

4. We implement multiple procedures to help us find the errors. Utilizing capacity increment clarity of a program. An enormous code is consistently hard to peruse. Breaking the code in littler Functions keeps the program sorted out, straightforward and makes it reusable.

Discussions

This project is a great platform because it allows students to perform their skills in assembly language by doing a game and they can develop their skill better while writing the code such as they can learn how to set text colour like we do in our project. Other than that, this project also requires a lot of discussions as we need to brainstorm ideas like what type of fruits, we are going to design by using notepad.

This project also can be fun if we can play with our friends as we can tease each other despite playing alone. In addition, we also applied a few skills that we learned in our coding and we also make an improvement by adding a new skill like ret and set text colour.

However, we also have weakness like we failed to form our flowchart. But, after being reprimanded by our lecturer, we managed to repair the fail flowchart. After doing second consultation, we also got comment about our project for not making it feels like a game and we also managed to fix it.

In the future, we will develop more skills to our project to make it more challenging and fun that can be played by all ages.

Conclusion

In conclusion, through this project we can enhance our skills and knowledge in the subject Computer Organization and Architecture. We also use the knowledge we have learned in the labs for this project. The smooth discussion among the group members for this project which led us to have better understanding among each other also contribute to the success on making this project. Nevertheless, we did enjoy the process of making this project and we hope that this project will help us in our future endeavours.

References

Iwatani, T. (3 June, 2020). *The 3 Primary Principles of Game Design*. Retrieved from Game Designing: https://www.gamedesigning.org/learn/game-design-principles/

Appendix

```
TITLE MASM TEAM_EAX                             (main.asm)

; Description:
;
; Revision date:

INCLUDE Irvine32.inc
.data

;The title
line1 byte "********** ********** **********    ********** **         ********** **
**",0
line2 byte "**      ** ********** **      **    **      ** **         ********** **
**",0
line3 byte "**      ** **         **      **    **      ** **         **          **
**",0
line4 byte "**      ** **         **      **    **      ** **         **          **
**",0
line5 byte "********** ********** **      **    ********** **         **********
****",0
line6 byte "**         ********** **********    **         **         **********
****",0
line7 byte "**         **         **       **   **         **         **          **
**",0
line8 byte "**         **         **        **  **         **         **          **
**",0
line9 byte "**         ********** **        **  **         ********** ********** **
**",0
line10 byte "**         ********** **        **  **         ********** ********** **
**",0

line11 byte "********** **********       **     ********** **********",0
line12 byte "********** **      **     **  **   ********** **********",0
line13 byte "**         **      **    **    **      **     **",0
line14 byte "**         **      **   **      **     **     **",0
line15 byte "**         **      **   **      **     **     **********",0
line16 byte "**         **********   **********     **     **********",0
line17 byte "**         **       **  **      **     **     **",0
line18 byte "**         **        ** **      **     **     **",0
line19 byte "********** **        ** **      **     **     **********",0
line20 byte "********** **        ** **      **     **     **********",0

;MAIN MENU
p2 byte "START GAME ( 1 )",0
p3 byte "HOW TO PLAY ( 2 )",0
P4 byte "EXIT ( 3 )",0
p5 byte "USER : ",0
p9 byte "======SCORE======",0
p7 byte "Your score : ",0
p8 byte "Total Score for this crate : ",0
p10 byte "======FINAL SCORE=====",0
p11 byte "crate 1 : ",0
p12 byte "crate 2 : ",0
p13 byte "crate 3 : ",0
```

```
;LUAR
ar1 byte "  \----------------------------------\",0
ar2 byte "   \                                  \        __",0
ar3 byte "    \                                  \       | \",0
ar4 byte "     >      THANK YOU FOR PLAYING       >------|  \       ______",0
ar5 byte "    /                                  /       --- \_____/**|_|_\____  |",0
ar6 byte "   /                                  /          \_______ --------- __>-}",0
ar7 byte "  /----------------------------------/              /  \_____|_____/   |",0
ar8 byte "                                                    *         |",0
ar9 byte "                                                             {O}",0
arr1 byte "       /*\       /*\       /*\       /*\       /*\       /*\       /*\",0
arr2 byte "      |***|     |***|     |***|     |***|     |***|     |***|     |***|",0
arr3 byte "       \*/       \*/ ____  \*/       \*/       \*/       \*/       \*/",0
arr4 byte "        |         |  |  |   |         |         |         |         |",0
arr5 byte "  ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^",0

;HOW TO PLAY
p6 byte "CHOOSE YOUR CRATE :)",0
ho byte "HOW TO PLAY",0
ho1 byte "1. CHOOSE ANY CRATE",0
ho2 byte "2. YOU HAVE THREE TIMES TO PICK A RANDOM FRUIT FORM THAT CRATE",0
ho3 byte "3. EACH FRUIT WILL TELL SOMETHING UNIQUE ABOUT YOU",0
ho4 byte "READY?  1 - YES / 2 - NO : ",0
ho5 byte "At the end of each game, there will be your score and your total score for that
crate",0
lu byte "THANK YOU FOR PLAYING :)",0

;CRATE
mula1 byte "CRATE 1 : THE EMOTIONS      CRATE 2 : FUTURE ",0
mula12 byte "|^^^^^^^^|                                |^^^^^^^^|",0
mula13 byte "|        |                                |        |",0
mula14 byte "|    1   |                                |    2   |",0
mula15 byte "|        |                                |        |",0
mula16 byte "|________|                                |________|",0
mula17 byte "CRATE 3 : ABOUT YOU",0
mula18 byte "|^^^^^^^^|",0
mula19 byte "|        |",0
mula20 byte "|    3   |",0
mula21 byte "|        |",0
mula22 byte "|________|",0
mula23 byte "CRATE NUMMBER : ",0

;USER PROMPT START GAME
kstart byte "PICK A FRUIT BY PRESSING ENTER",0
kdash byte "============================================",0
kpick1 byte "1st PICK : ",0
kpick2 byte "2nd PICK : ",0
kpick3 byte "3rd PICK : ",0
tung byte "WAITING ",0
tung1 byte "*",0

;array kotak
kotak dword 1,2,3
total dword 0,0,0

;DATA FOR FRUIT
fruit byte "  DURIAN",0
d1 byte "          ______",0
```

```
d2 byte "          |____|",0
d3 byte "            ||",0
d4 byte "          ^^^^^^",0
d5 byte "        ^>>>>>>>>^",0
d6 byte "      ^<<<<<<<<<<<<^",0
d7 byte "   ^^^^^^^^^^^^^^^^^^^^",0
d8 byte "  ^>>>>>>>>>>>>>>>>>>>>^",0
d9 byte "^<<<<<<<<<<<<<<<<<<<<<<<^",0
d10 byte "  ^^^^^^^^^^^^^^^^^^^^^",0
d11 byte "   ^>>>>>>>>>>>>>>>>>^",0
d12 byte "     ^<<<<<<<<<<<<<^",0
d13 byte "       ^^^^^^^^^^^",0
d14 byte "         ^>>>>>^",0

fruit2 byte " GRAPE",0
gp1 byte " __ {_/ ",0
gp2 byte " \_}\\ _",0
gp3 byte "    _\(_)_",0
gp4 byte "   (_)_)(_)_",0
gp5 byte "  (_)(_)_)(_)",0
gp6 byte "   (_)(_))_)",0
gp7 byte "    (_(_(_)",0
gp8 byte "     (_)_)",0
gp9 byte "      (_)",0

fruit3 byte " APPLE",0
ap1 byte "        __",0
ap2 byte "        \/.--,",0
ap3 byte "        //_.'",0
ap4 byte "   .-''-/''-.",0
ap5 byte "  /       __ \",0
ap6 byte " /        \\\ \",0
ap7 byte " |         || |",0
ap8 byte " \            /",0
ap9 byte " \  \         /",0
ap10 byte "  \  '-      /",0
ap11 byte "   '-.__.__.'",0

fruit4 byte " BANANA",0
bn1 byte " _",0
bn2 byte "//\",0
bn3 byte "V  \",0
bn4 byte " \  \_",0
bn5 byte "  \,'.`-.",0
bn6 byte "   |\ `. `.",0
bn7 byte "   ( \  `. `-.                        _,.-:\",0
bn8 byte "    \ \   `.  `-._             __..--' ,-';/",0
bn9 byte "     \ `.   `-.   `-..___..---'   _.--' ,'/",0
bn10 byte "      `. `.    `-._        __..--'    ,' /'",0
bn11 byte "        `. `-_     ``--..''       _.-' ,'",0
bn12 byte "          `-_ `-.___        __,--'   ,'",0
bn13 byte "             `-.__  `----''''    __.-'",0
bn14 byte "                  `--..____..--'",0

fruit5 byte " KIWI",0
o1 byte "    ___________________ ",0
o2 byte "   (_-_-_-_-_-_-_-_-_-_)...",0
o3 byte "  (    o \   |     /  o   ) .....",0
```

```
o4 byte " (   o        \  |  /     o) .....",0
o5 byte "(    o  \ **/   o      )......",0
o6 byte "( ~~~~~~~~  **** ~~~~~~~~ ).......",0
o7 byte "(       o  /****\    o      ).......",0
o8 byte " (   o        /  **        \     o ).......",0
o9 byte "  (  o/  o |       o \  o  )......",0
o10 byte "   (____/_____|____\___).....",0
o11 byte "    (_-_-_-_-_-_-_-_-_)....",0

fruit6 byte " MANGO",0
mg1 byte "      __",0
mg2 byte "      ||  ",0
mg3 byte "     _||_",0
mg4 byte "   'cccccc'",0
mg5 byte "  'cccccccc'",0
mg6 byte "  'ccccccccc'",0
mg7 byte "  'ccccccccc'",0
mg8 byte "   'cccccccc'",0
mg9 byte "    'cccccc'",0
mg10 byte "     'cccc'",0
mg11 byte "     'ccc'",0
mg12 byte "     'cc'",0
mg13 byte "     'c'",0

fruit7 byte " PINEAPPLE",0
pn1 byte "  \|/",0
pn2 byte "  AXA",0
pn3 byte " /XXX\",0
pn4 byte "/XXXXX\",0
pn5 byte "\XXXXX/",0
pn6 byte " \XXX/",0
pn7 byte "  '^'",0

fruit8 byte " NUT",0
nut1 byte " ,+.",0
nut2 byte "((|))",0
nut3 byte " )|(",0
nut4 byte "((|))",0
nut5 byte " `-'",0

fruit9 byte " STAR FRUIT",0
stt1 byte "                *",0
stt2 byte "               *^*",0
stt3 byte "              *^^^*",0
stt4 byte "             *^^^^^^*",0
stt5 byte "            *^^^^^^^^* ",0
stt6 byte " * * * * * *^^^^^^^^^^* * * * *",0
stt7 byte "  *^^^^^^^^^^^^^^^^^^^^^^^^^^*",0
stt8 byte "   *^^^^^^^^^^^^^^^^^^^^^^^^*",0
stt9 byte "    *^^^^^^^^^^^^^^^^^^^^^^*",0
stt10 byte "     *^^^^^^^^^^^^^^^^^^^^*",0
stt11 byte "      *^^^^^^^^^^^^^^^^^^*",0
stt12 byte "      *^^^^^^^^^*^^^^^^^^*",0
stt13 byte "     *^^^^^^^^* *^^^^^^^^*",0
stt14 byte "    *^^^^^^^^*   *^^^^^^^^*",0
stt15 byte "   *^^^^^^^^*     *^^^^^^^^*",0
stt16 byte "  *^^^^^^^^*       *^^^^^^^^*",0
stt17 byte " *^^^^^^^^*         ^^^^^^^^^*",0
```

```
stt18 byte "* * * * **           ** * * * *",0

fruit10 byte "       DRAGON FRUIT",0
dg1 byte "          _",0
dg2 byte "         //",0
dg3 byte "     _  || __",0
dg4 byte "  .--\\`||/.'--.",0
dg5 byte " / .-'.- +\`-._V`.",0
dg6 byte "JV `'.'/||//`-' V \",0
dg7 byte "| V V|/ |/ \| V V  L",0
dg8 byte "J VV  V  V  V   V V|",0
dg9 byte " L V  V V V VV V V F",0
dg10 byte " |V V V  V  V  VV /",0
dg11 byte " J  V   V V  V  V/",0
dg12 byte "  \V  V    V  V J",0
dg13 byte "   L V  V  V  V F",0
dg14 byte "   JV V  V  V V/",0
dg15 byte "    \ V    V .'",0
dg16 byte "     `-.V_.-'",0

;CRATE 1
des byte "The best and most beautiful things in the world cannot be seen, they must be
felt with the heart ",0
des12 byte "One thing you can't hide is when you're crippled inside",0
des13 byte "The emotion that can break your heart is sometimes the very one that heals
it",0
des14 byte "Your emotions are the slaves to your thoughts, and you are the slave to your
emotions",0
des15 byte "Pity those who don't feel anything at all because life is full of colors of
feelings",0
des16 byte "The world is a tragedy to those who feel, but a comedy to those who think",0
des17 byte "Happy, sad, fear, anger, disgust and surprise are all part and parcel of
life",0
des18 byte "I envy people that know love. That have someone who takes them as they are",0
des19 byte "Sometimes I think, I need a spare heart to feel all the things I feel",0
des2 byte "True love is not a hide and seek game, in true love, both lovers seek each
other",0

;CRATE 2
des21 byte "You will be rich",0
des22 byte "You will marry to your crush",0
des23 byte "You will die",0
des24 byte "You will ugly",0
des25 byte "You will success",0
des26 byte "You will be handsome or pretty",0
des27 byte "You will have small cottage",0
des28 byte "You will have your favourite car",0
des29 byte "You will marry to your enemy",0
des3 byte "You will be a good person",0

;CRATE 3
des31 byte "People can see you as stubborn and headstrong because you definitely have a
dominant personality",0
des32 byte "You have a lot of enthusiasm, but it fades rather quickly",0
des33 byte "Sometimes your emotions weigh you down, but you generally feel free from
them",0
des34 byte "You have the classic Type A personality",0
```

```
des35 byte "At times, you can be a bit too serious and you tend to put too much pressure
on yourself",0
des36 byte "You can't change how people treat you or what they say about you.All you can
do is change how you react to it.",0
des37 byte "Don't hold to anger,hurt or pain. They steal your energy and keep you from
love.",0
des38 byte "Don't give up on the person you are becoming.",0
des39 byte "Stop dreaming about your bucket list and start living it.",0
des4 byte "Love yourself like I love you.",0

;counter for each crate
cok11 dword 1
cok12 dword 1
cok13 dword 1

;ask prompt
last byte "WANT TO PLAY AGAIN (1 – YES , 2 – NO ) : ",0


.code
main PROC

       call colour1                  ;set the  initial colour for the text
       call ani                             ;proc for colour animation
       call prompt_2                 ;proc print start game
       call prompt_3                 ;proc print how to play
       call prompt_4                 ;proc print exit
       call Crlf
       call Crlf
       call prompt_5                 ;call user prompt
       call ReadDec                  ;input from user
       cmp eax,1                            ;if input 1
       je start
       cmp eax,2                            ;if input 2
       je how
       jmp luar                             ;if input 3
main ENDP

colour1 proc
       mov ebx,green+7
       mov eax,ebx
       call SetTextColor
       ret
colour1 endp

ani proc                                    ;animation proc
       mov ecx,4
L1:
       call maintitle                       ;proc print the PERPLEX CRATE title
       mov eax,900
       call Delay                           ;screen delay for 900 miliseconds
       call Clrscr
       add ebx,1
       mov eax,ebx
       call SetTextColor             ;set new text colour
       loop L1
ani endp
```

```
maintitle proc                              ;proc print the title
       mov edx,offset line1
       call Writestring
       call Crlf
       mov edx,offset line2
       call Writestring
       call Crlf
       mov edx,offset line3
       call Writestring
       call Crlf
       mov edx,offset line4
       call Writestring
       call Crlf
       mov edx,offset line5
       call Writestring
       call Crlf
       mov edx,offset line6
       call Writestring
       call Crlf
       mov edx,offset line7
       call Writestring
       call Crlf
       mov edx,offset line8
       call Writestring
       call Crlf
       mov edx,offset line9
       call Writestring
       call Crlf
       mov edx,offset line10
       call Writestring
       call Crlf
       call Crlf
       mov edx,offset line11
       call Writestring
       call Crlf
       mov edx,offset line12
       call Writestring
       call Crlf
       mov edx,offset line13
       call Writestring
       call Crlf
       mov edx,offset line14
       call Writestring
       call Crlf
       mov edx,offset line15
       call Writestring
       call Crlf
       mov edx,offset line16
       call Writestring
       call Crlf
       mov edx,offset line17
       call Writestring
       call Crlf
       mov edx,offset line18
       call Writestring
       call Crlf
       mov edx,offset line19
       call Writestring
```

```
        call Crlf
        mov edx,offset line20
        call Writestring
        call Crlf
        call Crlf
        ret
maintitle endp

prompt_2 proc                           ;proc print start game
        mov eax,green+13
        call SetTextColor
        mov edx,offset p2
        call Writestring
        call Crlf
        ret
prompt_2 endp

prompt_3 proc                           ;proc print how to play
call Crlf
        mov eax,green+12
        call SetTextColor
        mov edx,offset p3
        call Writestring
        call crlf
        ret
prompt_3 endp

prompt_4 proc                           ;proc print exit
call Crlf
        mov eax,green+7
        call SetTextColor
        mov edx,offset p4
        call Writestring
        call Crlf
        ret
prompt_4 endp

prompt_5 proc                           ;proc ask user
        mov eax,3
        call SetTextColor
        mov edx,offset p5
        call Writestring
        ret
prompt_5 endp

start proc                                     ;proc start game
        mov eax,300
        call Delay
        call Clrscr
        call crate                             ;call proc crate to display all crates
        ret
start endp

crate proc                                     ;proc display all crates
        mov eax,green+13                ;set text colour
        call SetTextColor
        call Clrscr
        mov edx,offset p6
        call Writestring
        call Crlf
        call Crlf
        mov edx,offset mula1
        call Writestring
        call Crlf
        mov edx,offset mula12
        call Writestring
        call Crlf
        mov edx,offset mula13
        call Writestring
        call Crlf
        mov edx,offset mula14
        call Writestring
        call Crlf
        mov edx,offset mula15
        call Writestring
        call Crlf
        mov edx,offset mula16
        call Writestring
        call Crlf
        call Crlf
        mov edx,offset mula17
        call Writestring
        call Crlf
        mov edx,offset mula18
        call Writestring
        call Crlf
        mov edx,offset mula19
        call Writestring
        call Crlf
        mov edx,offset mula20
        call Writestring
        call Crlf
        mov edx,offset mula21
        call Writestring
        call Crlf
        mov edx,offset mula22
        call Writestring
        call Crlf
        call Crlf
        mov edx,offset mula23
        call Writestring
        call ReadDec
        cmp eax,kotak
        je k1
        cmp eax,kotak+4
        je k2
        jmp k3
crate endp

how proc                                        ;proc display all how to play
instructions
        call Clrscr
        mov eax,green
        call SetTextColor
        mov eax,300
        call Delay
        mov edx,offset ho
        call Writestring
        call Crlf
        call Crlf
        mov edx,offset ho1
        call Writestring
        call Crlf
        mov edx,offset ho2
        call Writestring
        call Crlf
        mov edx,offset ho3
        call Writestring
        call Crlf
        call Crlf
        mov edx,offset ho5
        call Writestring
        call Crlf
        call Crlf
        mov edx,offset ho4
        call Writestring
        call ReadDec
        cmp eax,1
        je start
        jmp luar                                        ;jmp luar if user enter 2
how endp

luar proc                                               ;proc print "Thank you for Playing"
        call Clrscr
        mov edx,offset p10
        call Writestring
        call Crlf
        mov edi,offset total
        mov eax,[edi]
        mov edx,offset p11
        call Writestring
        call WriteDec
        call Crlf
        add edi,TYPE total
        mov eax,[edi]
        mov edx,offset p12
        call Writestring
        call WriteDec
        call Crlf
        add edi,TYPE total
        mov eax,[edi]
        mov edx,offset p13
        call Writestring
        call WriteDec
        call Crlf
        call Crlf
        mov edx,offset ar1
        call Writestring
        call Crlf
        mov edx,offset ar2
        call Writestring
        call Crlf
        mov edx,offset ar3
        call Writestring
        call Crlf
        mov edx,offset ar4
        call Writestring
        call Crlf
        mov edx,offset ar5
        call Writestring
        call Crlf
        mov edx,offset ar6
        call Writestring
        call Crlf
        mov edx,offset ar7
        call Writestring
        call Crlf
        mov edx,offset ar8
        call Writestring
        call Crlf
        mov edx,offset ar9
        call Writestring
        call Crlf
        call Crlf
        mov edx,offset arr1
        call Writestring
        call Crlf
        mov edx,offset arr2
        call Writestring
        call Crlf
        mov edx,offset arr3
        call Writestring
        call Crlf
        mov edx,offset arr4
        call Writestring
        call Crlf
        mov edx,offset arr5
        call Writestring
        call Crlf
        exit
luar endp

k1 proc                                         ;proc crate 1 if user choose crate
1
        call Clrscr
        mov edx,offset kstart
        call Writestring
        call Crlf
        call Crlf
        mov edx,offset kdash
        call Writestring
        call Crlf
        mov edx,offset kpick1
        call Writestring
        call ReadDec
        call tunggu
        call k11                        ;call k11 proc (random fruits and desc)

        inc cok11
        call Crlf
        call Crlf
        mov edx,offset kdash
        call Writestring
        call Crlf
        call Crlf
        mov edx,offset kpick2
        call Writestring
        call ReadDec
        call tunggu
        call k11                          ;call k11 proc (random fruits and desc)

        inc cok11
        call Crlf
        call Crlf
        mov edx,offset kdash
        call Writestring
        call Crlf
        call Crlf
        mov edx,offset kpick3
        call Writestring
        call ReadDec
        call tunggu
        call k11                          ;call k11 proc (random fruits and desc)

        call Crlf
        call Crlf
        mov edx,offset kdash
        call Writestring
        call Crlf
        call score1
        call ask                                 ;call ask proc
k1 endp

k2 proc                                                 ;proc crate 2 if user choose crate
2
        call Clrscr
        mov edx,offset kstart
        call Writestring
        call Crlf
        mov edx,offset kdash
        call Writestring
        call Crlf
        call Crlf
        mov edx,offset kpick1
        call Writestring
        call ReadDec
        call tunggu
        mov eax,cok12
        call k12                                 ;call k12 proc (random fruits and desc)

        inc cok12
        call Crlf
        call Crlf
        mov edx,offset kdash
        call Writestring
        call Crlf
        call Crlf
        mov edx,offset kpick2
        call Writestring
        call ReadDec
        call tunggu
        mov eax,cok12
        call k12                                ;call k12 proc (random fruits and desc)

        inc cok12
        call Crlf
        call Crlf
        mov edx,offset kdash
        call Writestring
        call Crlf
        call Crlf
        mov edx,offset kpick3
        call Writestring
        call ReadDec
        call tunggu
        mov eax,cok12
        call k12                                ;call k12 proc (random fruits and desc)

        call Crlf
        call Crlf
        mov edx,offset kdash
        call Writestring
        call Crlf
        mov edx,offset kdash
        call Writestring
        call Crlf
        call score2
        call ask                                ;call ask proc
k2 endp

k3 proc                                                 ;proc crate 3 if user choose crate
3
        call Clrscr
        mov edx,offset kstart
        call Writestring
        call Crlf
        mov edx,offset kdash
        call Writestring
        call Crlf
        call Crlf
        mov edx,offset kpick1
        call Writestring
        call ReadDec
        call tunggu
        mov eax,cok13
        call k13                                ;call k13 proc (random fruits and desc)

        inc cok13
        call Crlf
        call Crlf
        mov edx,offset kdash
        call Writestring
        call Crlf
        call Crlf
        mov edx,offset kpick2
        call Writestring
        call ReadDec
        call tunggu
        mov eax,cok13
        call k13                            ;call k13 proc (random fruits and desc)

        inc cok13
        call Crlf
        call Crlf
        mov edx,offset kdash
        call Writestring
        call Crlf
        call Crlf
        mov edx,offset kpick3
        call Writestring
        call ReadDec
        call tunggu
        mov eax,cok13
        call k13                            ;call k13 proc (random fruits and desc)

        call Crlf
        call Crlf
        mov edx,offset kdash
        call Writestring
        call Crlf
        mov edx,offset kdash
        call Writestring
        call Crlf
        call score3
        call ask                            ;call ask proc
k3 endp

k11 proc                                    ;proc looping for random fruits and
description for crate 1

        mov eax,cok11
        cmp eax,1d
        je L1
        cmp eax,2d
        je L2
        cmp eax,3d
        je L3
        cmp eax,4d
        je L4
        cmp eax,5d
        je L5
        cmp eax,6d
        je L6
        cmp eax,7d
        je L7
        cmp eax,8d
        je L8
        cmp eax,9d
        je L9
        cmp eax,10d
        je L10

L1:
        mov edx,offset fruit
        call Writestring
        call Crlf
        call Crlf
        call durian
        mov edx,offset des
        call Writestring
        ret
L2:
        mov edx,offset fruit2
        call Writestring
        call Crlf
        call Crlf
        call grape
        mov edx,offset des12
        call Writestring
        ret
L3:
        mov edx,offset fruit3
        call Writestring
        call Crlf
        call Crlf
        call apple
        mov edx,offset des13
        call Writestring
        ret
L4:
        mov edx,offset fruit4
        call Writestring
        call Crlf
        call Crlf
        call banana
        mov edx,offset des14
        call Writestring
        ret
L5:
        mov edx,offset fruit5
        call Writestring
        call Crlf
        call Crlf
        call kiwi
        mov edx,offset des15
        call Writestring
        call Crlf
        ret
L6:
        mov edx,offset fruit6
        call Writestring
        call Crlf
        call Crlf
        call mango
        mov edx,offset des16
        call Writestring
        call Crlf
        ret
L7:
        mov edx,offset fruit7
        call Writestring
        call Crlf
        call Crlf
        call pine
        mov edx,offset des17
        call Writestring
        call Crlf
        ret
L8:
        mov edx,offset fruit8
        call Writestring
        call Crlf
        call Crlf
        call kacang
        mov edx,offset des18
        call Writestring
        call Crlf
        ret
L9:
        mov edx,offset fruit9
        call Writestring
        call Crlf
        call Crlf
        call star
        mov edx,offset des19
        call Writestring
        call Crlf
        ret
L10:
        mov edx,offset fruit10
        call Writestring
        call Crlf
        call Crlf
        call dragon
        mov edx,offset des2
        call Writestring
        call Crlf
        ret
k11 endp

k12 proc                                          ;proc looping for random fruits and
description for crate 2

        cmp eax,1
        je L1
        cmp eax,2
        je L2
        cmp eax,3
        je L3
        cmp eax,4
        je L4
        cmp eax,5
        je L5
        cmp eax,6
        je L6
        cmp eax,7
        je L7
        cmp eax,8
        je L8
        cmp eax,9
        je L9
        cmp eax,10
        je L10

L1:
        mov edx,offset fruit10
        call Writestring
        call Crlf
        call Crlf
        call dragon
        mov edx,offset des21
        call Writestring
        ret
L2:
        mov edx,offset fruit9
        call Writestring
        call Crlf
        call Crlf
        call star
        mov edx,offset des22
        call Writestring
        ret
L3:
        mov edx,offset fruit8
        call Writestring
        call Crlf
        call Crlf
        call kacang
        mov edx,offset des23
        call Writestring
        ret
L4:
        mov edx,offset fruit7
        call Writestring
        call Crlf
        call Crlf
        call pine
        mov edx,offset des24
        call Writestring
        ret
L5:
        mov edx,offset fruit6
        call Writestring
        call Crlf
        call Crlf
        call mango
        mov edx,offset des25
        call Writestring
        call Crlf
        ret
L6:
        mov edx,offset fruit5
        call Writestring
        call Crlf
        call Crlf
        call kiwi
        mov edx,offset des26
        call Writestring
        ret
L7:
        mov edx,offset fruit4
        call Writestring
        call Crlf
        call Crlf
        call banana
        mov edx,offset des27
        call Writestring
        ret
L8:
        mov edx,offset fruit3
        call Writestring
        call Crlf
        call Crlf
        call apple
        mov edx,offset des28
        call Writestring
        ret
L9:
        mov edx,offset fruit2
        call Writestring
        call Crlf
        call Crlf
        call grape
        mov edx,offset des29
        call Writestring
        ret
L10:
        mov edx,offset fruit
        call Writestring
        call Crlf
        call Crlf
        call durian
        mov edx,offset des3
        call Writestring
        call Crlf
        ret

k12 endp

k13 proc                                        ;proc looping for random fruits and
description for crate 3

        cmp eax,1
        je L1
        cmp eax,2
        je L2
        cmp eax,3
        je L3
        cmp eax,4
        je L4
        cmp eax,5
        je L5
        cmp eax,6
        je L6
        cmp eax,7
        je L7
        cmp eax,8
        je L8
        cmp eax,9
        je L9
        cmp eax,10
        je L10

L1:
        mov edx,offset fruit9
        call Writestring
        call Crlf
        call Crlf
        call star
        mov edx,offset des31
        call Writestring
        ret
L2:
        mov edx,offset fruit4
        call Writestring
        call Crlf
        call Crlf
        call banana
        mov edx,offset des32
        call Writestring
        ret
L3:
        mov edx,offset fruit6
        call Writestring
        call Crlf
        call Crlf
        call mango
        mov edx,offset des33
        call Writestring
        ret
L4:
        mov edx,offset fruit10
        call Writestring
        call Crlf
        call Crlf
        call dragon
        mov edx,offset des34
        call Writestring
        ret
L5:
        mov edx,offset fruit7
        call Writestring
        call Crlf
        call Crlf
        call pine
        mov edx,offset des35
        call Writestring
        call Crlf
        ret
L6:
        mov edx,offset fruit2
        call Writestring
        call Crlf
        call Crlf
        call grape
        mov edx,offset des36
        call Writestring
        call Crlf
        ret
L7:
        mov edx,offset fruit8
        call Writestring
        call Crlf
        call Crlf
        call kacang
        mov edx,offset des37
        call Writestring
        call Crlf
        ret
L8:
        mov edx,offset fruit3
        call Writestring
        call Crlf
        call Crlf
        call apple
        mov edx,offset des38
        call Writestring
        call Crlf
        ret
L9:
        mov edx,offset fruit5
        call Writestring
        call Crlf
        call Crlf
        call kiwi
        mov edx,offset des39
        call Writestring
        call Crlf
        ret
L10:
        mov edx,offset fruit
        call Writestring
        call Crlf
        call Crlf
        call durian
        mov edx,offset des4
        call Writestring
        call Crlf
        ret


k13 endp

tunggu proc                                     ;proc print animation for waiting * * * *

        mov edx,offset tung
        call Writestring
        mov edx,offset tung1
        call Writestring
        mov eax,370
        call Delay
        call Writestring
        call Delay
        call Writestring
        call Delay
        call Writestring
        call Delay
        call Writestring
        call Delay
        call Crlf
        ret
tunggu endp

score1 proc

        call Crlf
        call Crlf
        call Randomize
        mov eax,100
        call RandomRange
        mov edi,offset total
        add [edi],eax
        mov edx,offset p9
        call Writestring
        call Crlf
        mov edx,offset p7
        call Writestring
        call WriteDec
        call Crlf
        mov edx,offset p8
        call Writestring
        mov eax,[edi]
        call WriteDec
        call Crlf
        ret

score1 endp

score2 proc

        call Crlf
        call Crlf
        call Randomize
        mov eax,100
        call RandomRange
        mov edi,offset total
        add edi,TYPE total
        add [edi],eax
        mov edx,offset p9
        call Writestring
        call Crlf
        mov edx,offset p7
        call Writestring
        call WriteDec
        call Crlf
        mov edx,offset p8
        call Writestring
        mov eax,[edi]
        call WriteDec
        call Crlf
        ret
```

```
score2 endp

score3 proc

       call Crlf
       call Crlf
       call Randomize
       mov eax,100
       call RandomRange
       mov edi,offset total
       add edi,8
       add [edi],eax
       mov edx,offset p9
       call Writestring
       call Crlf
       mov edx,offset p7
       call Writestring
       call WriteDec
       call Crlf
       mov edx,offset p8
       call Writestring
       mov eax,[edi]
       call WriteDec
       call Crlf
       ret

score3 endp

ask proc                                           ;proc ask will ask the user if want to
continnue play or not

       call Crlf
       mov eax,green+12
       call SetTextColor
       mov edx,offset last
       call Writestring
       call ReadDec
       cmp eax,1                                   ;continue play will jump to choose crate
       je crate
       cmp eax,2                                   ;exit jump to luar proc
       je luar

ask endp

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

kiwi proc

       mov eax,green+8
       call SetTextColor
       mov edx,offset o1
       call Writestring
       call Crlf
       mov edx,offset o2
       call Writestring
       call Crlf
       mov edx,offset o3
```

```
        call Writestring
        call Crlf
        mov edx,offset o4
        call Writestring
        call Crlf
        mov edx,offset o5
        call Writestring
        call Crlf
        mov edx,offset o6
        call Writestring
        call Crlf
        mov edx,offset o7
        call Writestring
        call Crlf
        mov edx,offset o8
        call Writestring
        call Crlf
        mov edx,offset o9
        call Writestring
        call Crlf
        mov edx,offset o10
        call Writestring
        call Crlf
        mov edx,offset o11
        call Writestring
        call Crlf
        call Crlf
        call Crlf
        mov eax,green+13
        call SetTextColor
        ret
kiwi endp

durian proc

        mov eax,green+12
        call SetTextColor
        mov edx,offset d1
        call Writestring
        call Crlf
        mov edx,offset d2
        call Writestring
        call Crlf
        mov edx,offset d3
        call Writestring
        call Crlf
        mov edx,offset d4
        call Writestring
        call Crlf
        mov edx,offset d5
        call Writestring
        call Crlf
        mov edx,offset d6
        call Writestring
        call Crlf
        mov edx,offset d7
        call Writestring
        call Crlf
        mov edx,offset d8
        call Writestring
        call Crlf
        mov edx,offset d9
        call Writestring
        call Crlf
        mov edx,offset d10
        call Writestring
        call Crlf
        mov edx,offset d11
        call Writestring
        call Crlf
        mov edx,offset d12
        call Writestring
        call Crlf
        mov edx,offset d13
        call Writestring
        call Crlf
        mov edx,offset d14
        call Writestring
        call Crlf
        call Crlf
        call Crlf
        mov eax,green+13
        call SetTextColor
        ret
durian endp

grape proc

        mov eax,green+11
        call SetTextColor
        mov edx,offset gp1
        call Writestring
        call Crlf
        mov edx,offset gp2
        call Writestring
        call Crlf
        mov edx,offset gp3
        call Writestring
        call Crlf
        mov edx,offset gp4
        call Writestring
        call Crlf
        mov edx,offset gp5
        call Writestring
        call Crlf
        mov edx,offset gp6
        call Writestring
        call Crlf
        mov edx,offset gp7
        call Writestring
        call Crlf
        mov edx,offset gp8
        call Writestring
        call Crlf
        mov edx,offset gp9
        call Writestring
        call Crlf
        call Crlf
        mov eax,green+13
        call SetTExtColor
        ret
grape endp

apple proc

        mov eax,green+10
        call SetTextColor
        mov edx,offset ap1
        call Writestring
        call Crlf
        mov edx,offset ap2
        call Writestring
        call Crlf
        mov edx,offset ap3
        call Writestring
        call Crlf
        mov edx,offset ap4
        call Writestring
        call Crlf
        mov edx,offset ap5
        call Writestring
        call Crlf
        mov edx,offset ap6
        call Writestring
        call Crlf
        mov edx,offset ap7
        call Writestring
        call Crlf
        mov edx,offset ap8
        call Writestring
        call Crlf
        mov edx,offset ap9
        call Writestring
        call Crlf
        mov edx,offset ap10
        call Writestring
        call Crlf
        mov edx,offset ap11
        call Writestring
        call Crlf
        call Crlf
        mov eax,green+13
        call SetTExtColor
        ret

apple endp

banana proc

        mov eax,green+12
        call SetTextColor
        mov edx,offset bn1
        call Writestring
        call Crlf
        mov edx,offset bn2
        call Writestring
        call Crlf
        mov edx,offset bn3
        call Writestring
        call Crlf
        mov edx,offset bn4
        call Writestring
        call Crlf
        mov edx,offset bn5
        call Writestring
        call Crlf
        mov edx,offset bn6
        call Writestring
        call Crlf
        mov edx,offset bn7
        call Writestring
        call Crlf
        mov edx,offset bn8
        call Writestring
        call Crlf
        mov edx,offset bn9
        call Writestring
        call Crlf
        mov edx,offset bn10
        call Writestring
        call Crlf
        mov edx,offset bn11
        call Writestring
        call Crlf
        mov edx,offset bn12
        call Writestring
        call Crlf
        mov edx,offset bn13
        call Writestring
        call Crlf
        mov edx,offset bn14
        call Writestring
        call Crlf
        call Crlf
        mov eax,green+13
        call SetTextColor
        ret

banana endp

mango proc

        mov eax,green+12
        call SetTextColor
        mov edx,offset mg1
        call Writestring
        call Crlf
        mov edx,offset mg2
        call Writestring
        call Crlf
        mov edx,offset mg3
        call Writestring
        call Crlf
        mov edx,offset mg4
        call Writestring
        call Crlf
        mov edx,offset mg5
        call Writestring
        call Crlf
        mov edx,offset mg6
        call Writestring
        call Crlf
        mov edx,offset mg7
        call Writestring
        call Crlf
        mov edx,offset mg8
        call Writestring
        call Crlf
        mov edx,offset mg9
        call Writestring
        call Crlf
        mov edx,offset mg10
        call Writestring
        call Crlf
        mov edx,offset mg11
        call Writestring
        call Crlf
        mov edx,offset mg12
        call Writestring
        call Crlf
        mov edx,offset mg13
        call Writestring
        call Crlf
        call Crlf
        mov eax,green+13
        call SetTextColor
        ret

mango endp

pine proc

        mov eax,green
        call SetTextColor
        mov edx,offset pn1
        call Writestring
        call Crlf
        mov edx,offset pn2
        call Writestring
        call Crlf
        mov edx,offset pn3
        call Writestring
        call Crlf
        mov edx,offset pn4
        call Writestring
        call Crlf
        mov edx,offset pn5
        call Writestring
        call Crlf
        mov edx,offset pn6
        call Writestring
        call Crlf
        mov edx,offset pn7
        call Writestring
        call Crlf
        call Crlf
        mov eax,green+13
        call SetTextColor
        ret

pine endp

star proc

        mov eax,green
        call SetTextColor
        mov edx,offset stt1
        call Writestring
        call Crlf
        mov edx,offset stt2
        call Writestring
        call Crlf
        mov edx,offset stt3
        call Writestring
        call Crlf
        mov edx,offset stt4
        call Writestring
        call Crlf
        mov edx,offset stt5
        call Writestring
        call Crlf
        mov edx,offset stt6
        call Writestring
        call Crlf
        mov edx,offset stt7
        call Writestring
        call Crlf
        mov edx,offset stt8
        call Writestring
        call Crlf
        mov edx,offset stt9
        call Writestring
        call Crlf
        mov edx,offset stt10
        call Writestring
        call Crlf
        mov edx,offset stt11
        call Writestring
        call Crlf
        mov edx,offset stt12
        call Writestring
        call Crlf
        mov edx,offset stt13
        call Writestring
        call Crlf
        mov edx,offset stt14
        call Writestring
        call Crlf
        mov edx,offset stt15
        call Writestring
        call Crlf
        mov edx,offset stt16
        call Writestring
        call Crlf
        mov edx,offset stt17
        call Writestring
        call Crlf
        mov edx,offset stt18
        call Writestring
        call Crlf
        call Crlf
        mov eax,green+13
        call SetTExtColor
        ret

star endp

dragon proc

        mov eax,green+11
        call SetTextColor
        mov edx,offset dg1
        call Writestring
        call Crlf
        mov edx,offset dg2
        call Writestring
        call Crlf
        mov edx,offset dg3
        call Writestring
        call Crlf
        mov edx,offset dg4
        call Writestring
        call Crlf
        mov edx,offset dg5
        call Writestring
        call Crlf
        mov edx,offset dg6
        call Writestring
        call Crlf
        mov edx,offset dg7
        call Writestring
        call Crlf
        mov edx,offset dg8
        call Writestring
        call Crlf
        mov edx,offset dg9
        call Writestring
        call Crlf
        mov edx,offset dg10
        call Writestring
        call Crlf
        mov edx,offset dg11
        call Writestring
        call Crlf
        mov edx,offset dg12
        call Writestring
        call Crlf
        mov edx,offset dg13
        call Writestring
        call Crlf
        mov edx,offset dg14
        call Writestring
        call Crlf
        mov edx,offset dg15
        call Writestring
        call Crlf
        mov edx,offset dg16
        call Writestring
        call Crlf
        call Crlf
        mov eax,green+13
        call SetTextColor
        ret
dragon endp

kacang proc

        mov eax,green+8
        call SetTextColor
        mov edx,offset nut1
        call Writestring
        call Crlf
        mov edx,offset nut2
        call Writestring
        call Crlf
        mov edx,offset nut3
        call Writestring
        call Crlf
        mov edx,offset nut4
        call Writestring
        call Crlf
        mov edx,offset nut5
        call Writestring
        call Crlf
        call Crlf
        mov eax,green+13
        call SetTextColor
        ret

kacang endp
END main
```