



# กินหัวโผล่หาง

## Snake Adventure

กระบวนวิชา การเขียนโปรแกรมเชิงโครงสร้าง  
Structured Programming (204112)

นำเสนอ

ผู้ช่วยศาสตราจารย์ ดร.วัชร จ่าปามูล

อาจารย์ ดร.รัศมีทิพย์ วิตา

อาจารย์ กิตติพิชญ์ คุปตะวาณิช

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยเชียงใหม่



กินหัวโผล่หาง

## Snake Adventure

กระบวนวิชา

การเขียนโปรแกรมเชิงโครงสร้าง

STRUCTURED PROGRAMMING (204112)

นำเสนอต่อ

ผู้ช่วยศาสตราจารย์ ดร.วัชร จำปามูล

อาจารย์ ดร.รัศมีทิพย์ วิตา

อาจารย์ กิตติพิชญ์ คุปตะวาณิช

ภาคเรียนที่ 2 ปีการศึกษา 2558

ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์

มหาวิทยาลัยเชียงใหม่

## สมาชิก

>..... นายอริวัฒน์      คงสินสุไพศาล รหัสนักศึกษา 570510706  
    ออกแบบโปรแกรม  
    เขียนโปรแกรม  
    ทดสอบโปรแกรม

>..... นางสาวอังคณา      มวลใหม่                      รหัสนักศึกษา 570510709  
    ออกแบบโปรแกรม  
    ทดสอบโปรแกรม  
    รวบรวมรูปเล่มรายงาน

>..... นายชาญวุฒิ      ปุณลิต                      รหัสนักศึกษา 580510589  
    ออกแบบโปรแกรม  
    เขียนโปรแกรม  
    ทดสอบโปรแกรม

# คำนำ

รายงานฉบับนี้เป็นส่วนหนึ่งของกระบวนการวิชา การเขียนโปรแกรมเชิงโครงสร้าง  
[ Structured Programming (204112) ] โดยมีวัตถุประสงค์ให้นักศึกษาได้มีการค้นคว้า  
ศึกษาและทำความเข้าใจเกี่ยวกับการเขียนโปรแกรมคอมพิวเตอร์ ทั้งนี้นักศึกษาจะได้มีการ  
พัฒนาทักษะ ทั้งในด้านการออกแบบ พัฒนาโปรแกรม และการแก้ไขปัญหา

ทางคณะผู้จัดทำมีความยินดีอย่างยิ่งที่เขียนโปรแกรมนี้นี้ขึ้นมา ซึ่งทางผู้จัดทำได้  
คาดหวังเป็นอย่าง ยิ่งว่าโปรแกรมที่ได้จัดทำขึ้นมานั้นจะเป็นประโยชน์แก่ผู้ที่สนใจการเขียน  
โปรแกรม หากรายงานฉบับนี้มีข้อผิดพลาดประการใด ทางคณะผู้จัดทำขออภัย ณ ที่นี้ด้วย

คณะผู้จัดทำ

## หลักการและเหตุผล

ทุกคนบนโลกใบนี้ย่อมมีความเครียด ไม่ว่าจะเป็นความเครียดที่เกิดจากการทำงาน ความเครียดที่เกิดจากการทำการบ้าน หรือสิ่งอื่นๆ และจากที่กล่าวมาทำให้กลุ่มของพวกเราเลือกที่จะพัฒนาโปรแกรมที่ใช้แล้วสามารถช่วยลดความเครียดลงได้บ้าง จึงเกิดความคิดที่จะพัฒนาโปรแกรมเกี่ยวกับเกมง่ายๆ ที่คนทุกเพศ ทุกวัย สามารถเล่นร่วมกันได้ แล้วช่วยลดความเครียดได้ด้วย จึงเกิดการพัฒนาโปรแกรมเกม "กินหัวโผล่หาง" ขึ้นมา

## โครงการนี้จะช่วยแก้ปัญหาความเครียดได้อย่างไร ?

เมื่อความเครียดเกิดขึ้นมาพวกเราเชื่อว่าทุกคนย่อมต้องการหาวิธีทางเพื่อผ่อนคลายความเครียดให้กับตัวเอง และคนส่วนใหญ่มักจะแก้ปัญหาความเครียดโดยการเล่นเกมง่ายๆ เพื่อผ่อนคลายความเครียดที่เกิดขึ้น พวกเราจึงมีความคิดที่จะสร้างเกมนี้เพื่อตอบสนองให้กับผู้ที่ต้องการผ่อนคลายความเครียด โดยเกมนี้สามารถสร้างความเพลิดเพลินและความสนุกสนานให้กับผู้เล่นได้ในเวลาเดียวกัน

## ผลที่คาดว่าจะได้รับ

1. มีทักษะในการเขียนโปรแกรมเชิงประยุกต์มากยิ่งขึ้น
2. เข้าใจหลักการทำงานของฟังก์ชันพื้นฐานต่างๆซึ่งเกี่ยวข้องกับความรู้ในห้องเรียนมากยิ่งขึ้น
3. ได้พัฒนาหลักการคิดที่เป็นระบบในการทำงาน
4. ได้พัฒนาทักษะในการทำงานเป็นกลุ่ม

## ข้อมูลนำเข้า

- นำเข้าข้อมูลการเคลื่อนที่ของตัวงูจากผู้ผ่านการกดแป้นลูกศร ขึ้น, ลง, ซ้าย, และขวา
- ผู้ใช้งานจะเป็นผู้ควบคุมการเคลื่อนที่ของตัวงูเพื่อให้ไปตามเส้นทางที่ต้องการ

## การประมวลผลเกม

- เมื่อโปรแกรมได้รับคำสั่งจากผู้ใช้งาน ก็จะประมวลผลให้งูเดินตามการกดแป้นลูกศรเพื่อไปกินอาหารที่วางไว้ในแต่ละจุด
- หลังจากที่ได้กินอาหารแล้ว จะทำให้ตัวงูยาวขึ้นเรื่อยๆ และทุกครั้งที่ได้กินอาหารได้ จะทำให้แต้มคะแนนเพิ่มขึ้นทีละ 1 คะแนน
- เมื่องูกินอาหารได้ตามจำนวนที่ตั้งไว้ จะทำให้ระดับความเร็วของตัวงูเพิ่มขึ้นเรื่อยๆ
- ถ้าตัวงูเดินไปชนกับขอบกำแพง หรืองูเดินไปทับกับตัวเอง จะทำให้งูตายและเสียจำนวนการเล่นไป 1 ครั้ง และเมื่อเสียจำนวนการเล่นครบ 3 ครั้งแล้ว จะทำให้เกมสิ้นสุดทันที

## ข้อมูลส่งออก

- โปรแกรมจะทำการบันทึกชื่อและคะแนนของผู้เล่นไว้ในไฟล์ และสามารถเรียกดูสถิติของผู้เล่นได้ตลอด

## Source Code

```
#include <stdio.h>

#include <time.h>

#include <stdlib.h>

#include <conio.h>

#include <ctype.h>

#include <windows.h>

#include <process.h>


#define UP 72

#define DOWN 80

#define LEFT 75

#define RIGHT 77


int length;

int bend_no;

int len;

char key;

void record();

void load();

int life;

void Delay(long double);

void Move();

void Food();

int Score();

void Print();
```

```

void gotoxy(int x, int y);

void GotoXY(int x,int y);

void Bend();

void Boarder();

void Down();

void Left();

void Up();

void Right();

void ExitGame();

int Scoreonly();


struct coordinate{

    int x;

    int y;

    int direction;

};


typedef struct coordinate coordinate;

coordinate head, bend[500], food, body[30];

// bend ใช้บอกความยาวของตัวงู, body ใช้บันทึกจำนวนการเคลื่อนไหว

int main()

{   char key;

    Print();

    system("cls");

    load();

    length=5;

```



```
    head.x=25;

    head.y=20;

    head.direction=RIGHT;

    Boarder();

    Food();

    Life = 3; // จำนวนชีวิตในเกม

    bend[0]=head;

    Move();

    return 0;
}

void Move()
{   int a,i;

        do{

            Food();

            fflush(stdin);

            len=0;

            for(i=0;i<30;i++)

            {

                body[i].x=0;

                body[i].y=0;

                if(i==length)

                    break;

            }

            Delay(length);

            Boarder();
```

```

        if(head.direction==RIGHT)

            Right();

        else if(head.direction==LEFT)

            Left();

        else if(head.direction==DOWN)

            Down();

        else if(head.direction==UP)

            Up();

        ExitGame();
    } while(!kbhit());

```

```

a = getch();

if(a==27)

{

    system("cls");

    exit(0);

}

key=getch();

```

```

if((key==RIGHT&&head.direction!=LEFT&&head.direction!=RIGHT)|| (key
==LEFT&&head.direction!=RIGHT&&head.direction!=LEFT)|| (key==UP&&he
ad.direction!=DOWN&&head.direction!=UP)|| (key==DOWN&&head.directio
n!=UP&&head.direction!=DOWN)) {

    bend_no++;

    bend[bend_no]=head;

    head.direction=key;

```

```
        if(key==UP)
            head.y--;
        if(key==DOWN)
            head.y++;
        if(key==RIGHT)
            head.x++;
        if(key==LEFT)
            head.x--;
        Move();
    }
    else if(key==27)
    {
        system("cls");
        exit(0);
    }
    else
    {
        printf("\a");
        Move();
    }
}
```

```
void GotoXY(int x, int y)
{
    HANDLE a;

    COORD b;

    fflush(stdout);

    b.X = x;

    b.Y = y;

    a = GetStdHandle(STD_OUTPUT_HANDLE);

    SetConsoleCursorPosition(a,b);
}
```

```
void gotoxy(int x, int y)
{
    COORD coord;

    coord.X = x;

    coord.Y = y;

    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coord);
}
```

ฟังก์ชันการเคลื่อนที่ลงด้านล่าง: **void Down()**

```
void Down()
{
    int i;
    for(i=0;i<=(head.y-bend[bend_no].y)&&len<length;i++)
    {
        GotoXY(head.x,head.y-i);
        {
            if(len==0)
                printf("W");
            else
                printf("o");
        }
        body[len].x=head.x;
        body[len].y=head.y-i;
        len++;
    }
    Bend();
    if(!kbhit())
        head.y++;
}
```

ฟังก์ชันการเคลื่อนที่ขึ้นข้างบน:**void Up()**

```
void Up()
{
    int i;
    for(i=0;i<=(bend[bend_no].y-head.y)&&len<length;i++)
    {
        GotoXY(head.x,head.y+i);
        {
            if(len==0)
                printf("M");
            else
                printf("o");
        }
        body[len].x=head.x;
        body[len].y=head.y+i;
        len++;
    }
    Bend();
    if(!kbhit())
        head.y--;
}
```

ฟังก์ชันการเคลื่อนที่ไปทางขวา: **void Right()**

```
void Right()
{
    int i;
    for(i=0;i<=(head.x-bend[bend_no].x)&&len<length;i++)
    {
        GotoXY((head.x-i),head.y);
        body[len].x=head.x-i;
        body[len].y=head.y;
        GotoXY(body[len].x,body[len].y);
        {
            if(len==0)
                printf("X");
            else
                printf("o");
        }
        body[len].x=head.x-i;
        body[len].y=head.y;
        len++;
    }
    Bend();
    if(!kbhit())
        head.x++;
}
```

ฟังก์ชันการเคลื่อนที่ไปทางซ้าย :void Left()

```
void Left()
{
    int i;
    for(i=0;i<=(bend[bend_no].x-head.x)&&len<length;i++)
    {
        GotoXY((head.x+i),head.y);
        {
            if(len==0)
                printf("X");
            else
                printf("o");
        }
        body[len].x=head.x+i;
        body[len].y=head.y;
        len++;
    }
    Bend();
    if(!kbhit())
        head.x--;
}

void ExitGame()
{
    int i,check=0;
```



```

        for(i=4;i<length;i++)    //starts with 4 because it needs
        minimum 4 element to touch its own body

        {

            if(body[0].x==body[i].x&&body[0].y==body[i].y)

            {

                Check;

            }

            if(i==length||check!=0)

                break;

        }

        if(head.x<=10||head.x>=70||head.y<=10||head.y>=30||check!=0)

        {

            life--;

            if(life>=0)

            {

                head.x=25;

                head.y=20;

                bend_no=0;

                head.direction=RIGHT;

                Move();

            }

        else

        {

            system("cls");

            printf("All lives completed\nPress any key to quit the
game\n");

            record();

```

```

        exit(0);
    }
}
}

```

ฟังก์ชันการเพิ่มความยาวของตัวงู : **void Bend()**

```

void Bend()
{
    int i,j,diff;
    for(i=bend_no;i>=0&&len<length;i--)
    {
        if(bend[i].x==bend[i-1].x)
        {
            diff=bend[i].y-bend[i-1].y;
            if(diff<0)
            for(j=1;j<=(-diff);j++)
            {
                body[len].x=bend[i].x;
                body[len].y=bend[i].y+j;
                GotoXY(body[len].x,body[len].y);
                printf("o");
                len++;
                if(len==length)
                    break;
            }
        }
    }
}

```

```

        }
    else if(diff>0)
        for(j=1;j<=diff;j++)
        {
            body[len].x=bend[i].x;
            body[len].y=bend[i].y-j;
            GotoXY(body[len].x,body[len].y);
            printf("o");
            len++;
            if(len==length)
                break;
        }
    }
else if(bend[i].y==bend[i-1].y)
{
    diff=bend[i].x-bend[i-1].x;
    if(diff<0)
        for(j=1;j<=(-diff)&&len<length;j++)
        {
            body[len].x=bend[i].x+j;
            body[len].y=bend[i].y;
            GotoXY(body[len].x,body[len].y);
            printf("o");
            len++;
            if(len==length)
                break;
        }
    }
}

```

```
    } else if(diff>0)
    for(j=1;j<=diff&&len<length;j++)
    {
        body[len].x=bend[i].x-j;
        body[len].y=bend[i].y;
        GotoXY(body[len].x,body[len].y);
        printf("o");
        len++;
        if(len==length)
            break;
    }
}
```

```
}
```

ฟังก์ชันการโหลดเกม: **void**

```
void load(){
    int row,col;
    int r,q;
    gotoxy(36,14);
    printf("Loading...");
    gotoxy(30,15);
    for(r=1;r<=20;r++){
        for(q=0;q<=100000000;q++); //to display the character slowly
        printf("%c",177); // ตัวอักษรพิเศษจาก ASCII
    }
    gotoxy(40,16);
    printf("\nPlease press any key...");
    getch();
}
```

ฟังก์ชันกำหนดความเร็วการเคลื่อนที่ของงู :void Delay(long double k)

```
void Delay(long double k)
{
    Score();
    long double i;
    if(length >=5 && length <=10){
        for(i=0;i<=(500000000);i++);
    }else if(length > 10 && length <=20){
        for(i=0;i<=(250000000);i++);
    }
    else if(length > 20){
        for(i=0;i<=(125000000);i++);
    }
}
```

ฟังก์ชันการเคลื่อนที่ลงด้านล่าง: void

```
void Down()
{
    int i;
    body[len].x=head.x;
    body[len].y=head.y-i;
    len++;
}
Bend();
```

```
        if(!kbhit())
            head.y++;

    }
```

ฟังก์ชันการเคลื่อนที่ขึ้นข้างบน:**void**

```
void Up()
{
    int i;

    body[len].x=head.x;

    body[len].y=head.y+i;

    len++;

}

Bend();

if(!kbhit())

    head.y--;

}
```

ฟังก์ชันการเคลื่อนที่ไปทางขวา: void

```
void Right()
{
    int i;
        body[len].x=head.x-i;
        body[len].y=head.y;
        len++;
    }
    Bend();
    if(!kbhit())
        head.x++;
}
```

ฟังก์ชันการเคลื่อนที่ไปทางซ้าย :void

```
void Left()
{
    int i;
        body[len].x=head.x+i;
        body[len].y=head.y;
        len++;
    }
    Bend();
    if(!kbhit())
        head.x--;
}
```



ฟังก์ชันสร้างอาหาร :void

```
void Food()
{
    GotoXY(food.x, food.y);
    printf("+");
    if(head.x==food.x&&head.y==food.y)
    {
        length++;
        time_t a;
        a=time(0);
        srand(a);
        food.x=rand()%70;
        if(food.x<=10)
            food.x+=11;
        food.y=rand()%30;
        if(food.y<=10)
            food.y+=11;
    }
    else if(food.x==0)
    {
        food.x=rand()%70;
        if(food.x<=10)
            food.x+=11;
        food.y=rand()%30;
        if(food.y<=10)
```

```
        food.y+=11;

    }

}

void Boarder()
{
    system("cls");
    int i;
    for(i=10;i<71;i++)
    {
        GotoXY(i,10);
        printf("=");
        GotoXY(i,30);
        printf("=");
    }
    for(i=10;i<31;i++)
    {
        GotoXY(10,i);
        printf("|");
        GotoXY(70,i);
        printf("|");
    }
}
```

```

void Print()
{
    printf("\tWelcome to the Snake game.(press any key to
continue)\n");

    getch();

    system("cls");

    printf("\tGame instructions:\n");

    printf("\n-> Use arrow keys to move the snake.\n\n-> You will be
provided foods at the several coordinates of the screen which you have
to eat. Everytime you eat a food the length of the snake will be
increased by 1 element and thus the score.\n\n-> Here you are provided
with three lives. Your life will decrease as you hit the wall or snake's
body.\n\n-> You can pause by pressing any key. To continue press any
other key once again\n\n-> If you want to exit press esc. \n");

    printf("\n\n Press any button to play game...");

    if(getch() == 27)

    exit(0);
}

```

ฟังก์ชันบันทึกสถิติลงแฟ้มข้อมูล: void

```

void record(){

    char plname[20],nplname[20],cha,c;

    int i,j,px;

    FILE *info;

    info=fopen("record.txt","a+");

    getch();

    system("cls");

    printf("Enter your name :");

```

```

scanf("%[^\\n]",plname);

for(j=0;plname[j]!='\\0';j++){
    nplname[0]=toupper(plname[0]);
    if(plname[j-1]==' '){
        nplname[j]=toupper(plname[j]);
        nplname[j-1]=plname[j-1];}
    else nplname[j]=plname[j];
}

nplname[j]='\\0';

fprintf(info,"Player Name :%s \\n",nplname);

time_t mytime;

mytime = time(NULL);

fprintf(info,"Played Date:%s",ctime(&mytime));

fprintf(info,"Score:%d\\n",px=Scoreonly()); // เรียกดูคะแนน

fprintf(info,"\\nLevel:%d\\n",10); // เรียกดูระดับของเกม

for(i=0;i<=50;i++)

fprintf(info,"%c",'_');

fprintf(info,"\\n");

fclose(info);

printf("Do you want to see records ? press 'y'\\n");

cha=getch();

system("cls");

if(ch=='y'){

    info = fopen("record.txt","r");

```

```
do{
    putchar(c=getc(info));
}while(c!=EOF);}

fclose(info);
}

int Score()
{
    int score;
    GotoXY(20,8);
    score=length-5;
    printf("SCORE : %d", (length-5));
    score=length-5;
    GotoXY(50,8);
    printf("Life : %d", life);
    return score;
}

int Scoreonly()
{
    int score=Score();
    system("cls");
    return score;
}
```

## ฟังก์ชันในเกม

**void record();** ใช้สำหรับการบันทึกสถิติการเล่นเกมของผู้เล่นโดยจะบันทึกคะแนนลงเพิ่มข้อมูลโดยอัตโนมัติหลังจบเกม

**void load();** ใช้สำหรับหน้าจอขณะรอโหลดเข้าเกม

**void Delay();** ใช้สำหรับกำหนดความเร็วของการเคลื่อนที่ของตัวงูโดยที่ถ้างูกินอาหารครบ 6 ครั้งความเร็วจะเพิ่มขึ้น 1 ระดับ และความเร็วจะเพิ่มขึ้นอีกครั้งเมื่องูกินอาหารครบ 20 ครั้ง

**void Move();** ใช้สำหรับรับคำสั่งการเคลื่อนที่จากแป้นพิมพ์โดยการกดปุ่มลูกศรขึ้น ลง ซ้าย และขวา

**void Food();** ใช้สำหรับสุ่มสร้างอาหารของงูภายในขอบเขตของจอเกม โดยจะสร้างอาหารไม่ซ้ำที่เดิม

**void gotoxy(int x, int y);** ใช้กำหนดตำแหน่งการเคลื่อนที่ของตัวงู ซึ่ง x คือตำแหน่งแนวนอน และ y คือตำแหน่งแนวตั้ง โดยฟังก์ชันนี้จะทำงานสัมพันธ์กับการกดแป้นพิมพ์เพื่อสั่งให้ตัวงูเคลื่อนที่

**void Boarder();** ใช้สำหรับสร้างขอบเขตของหน้าจอเกม

**void Down(), void Up();** ใช้สำหรับคำสั่งเคลื่อนที่ขึ้น และ ลง

**void Left();** ใช้สำหรับคำสั่งเคลื่อนที่ไปทางซ้าย

**void Right();** ใช้สำหรับคำสั่งเคลื่อนที่ไปทางขวา

**void Bend();** ใช้สำหรับกำหนดความยาวของตัวงู โดยจะเพิ่มขึ้นหลังจากที่งูได้กินอาหารเข้าไปแล้ว

## Library ที่กำหนดเพิ่มเติม

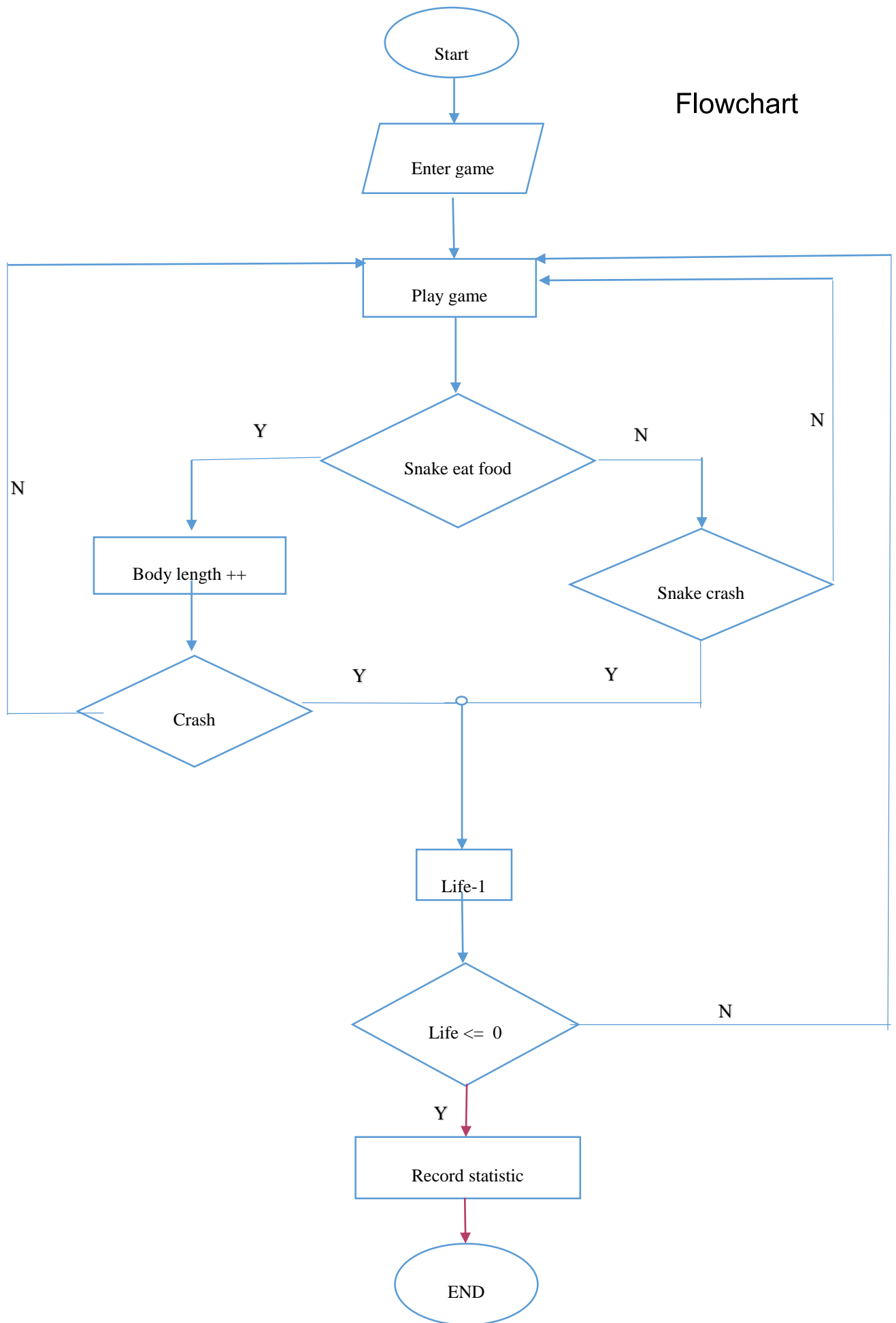
**Conio.h** : ใช้สำหรับ kbhit(); , gotoxy(); , getch(); and clrscr();

**Ctype.h** : เป็น library ที่ใช้สำหรับการจัดการตัวอักษร ซึ่งภายในโปรแกรมนี้ได้เลือกใช้ฟังก์ชัน toupper(); เพื่อให้ตัวอักษรทุกตัวเปลี่ยนเป็นตัวอักษรพิมพ์ใหญ่

**Time.h** : ใช้สำหรับ rand();

**Windows.h** : ใช้สำหรับการสร้าง GUI ของเกม

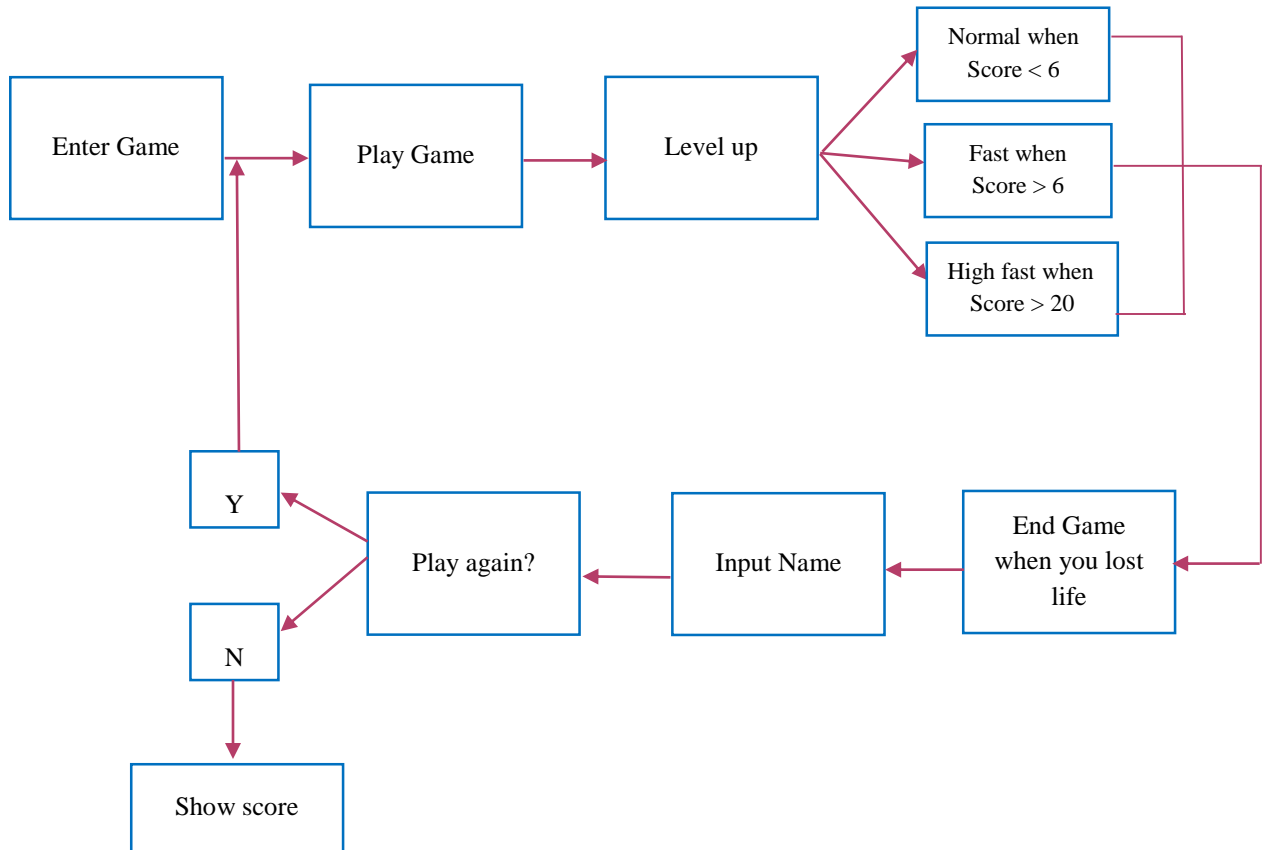
## Flowchart





## ขอบเขตของโครงการ

ขอบเขตทางสถาปัตยกรรม :



1. ใช้ภาษา C ในการเขียนโปรแกรม
2. ใช้โปรแกรม Sublime Text และ CodeBlock ร่วมกันในการเขียนโปรแกรม
3. ใช้ GCC Compiler และ MinGW เป็นตัว Build โปรแกรม
4. โปรแกรมนี้ใช้ได้ทั้งบน PC และ Notebook ที่มีระบบการทำงานวินโดวส์

**ขอบเขตของระบบงาน :**

ภายในโปรแกรมจะมีฟังก์ชันต่างๆดังนี้

ฟังก์ชันควบคุม :

1. บังคับขยับขึ้นบน - ลงล่าง
2. บังคับขยับเลี้ยวซ้าย - เลี้ยวขวา

ฟังก์ชันสิ่งกีดขวาง :

1. กำหนดจุดกีดขวางรูปแบบสุ่ม
2. กำหนดขอบเขตของสนาม

ฟังก์ชันวางอาหารแบบสุ่ม

ฟังก์ชันกำหนดความเร็วและความหน่วงของขู

ฟังก์ชันแสดงผลโปรแกรม

ฟังก์ชันอื่นๆที่อาจจะเพิ่มเติมเพื่อแก้ปัญหาภายหลัง

**ขอบเขตของข้อมูล :**

1. ผู้ใช้งานจะต้องเป็นผู้บังคับทิศทาง การเคลื่อนไหวของตัวขูเพื่อหลบหลีกสิ่งกีดขวาง และ เพื่อกินอาหารที่วางไว้ให้
2. เมื่อจบเกมให้โปรแกรมแสดงคะแนนที่ได้ พร้อมกับให้ผู้ใช้เลือกว่าจะเล่นต่อหรือออกจากเกม

### วิธีการดำเนินงาน:

1. ปรึกษาหารือเกี่ยวกับเรื่องที่จะทำกันโดยเสนอความคิดเห็นของแต่ละคนเพื่อนำมาปรับใช้กับงานที่ทำกัน
2. แบ่งงานโดยแบ่งตามความเหมาะสมของแต่ละบุคคล
3. ปรึกษากันเกี่ยวกับปัญหาที่เกิดขึ้นระหว่างการทำงาน
4. ปรึกษาอาจารย์เพื่อนหรือรุ่นพี่เพื่อขอความช่วยเหลือในเรื่องที่แก้ไขไม่ได้
5. สรุปและรวบรวมงานที่แต่ละคนทำและนำมาเสนอ อธิบายเกี่ยวกับงานที่ได้รับมอบหมาย

### วิธีการศึกษาค้นคว้า:

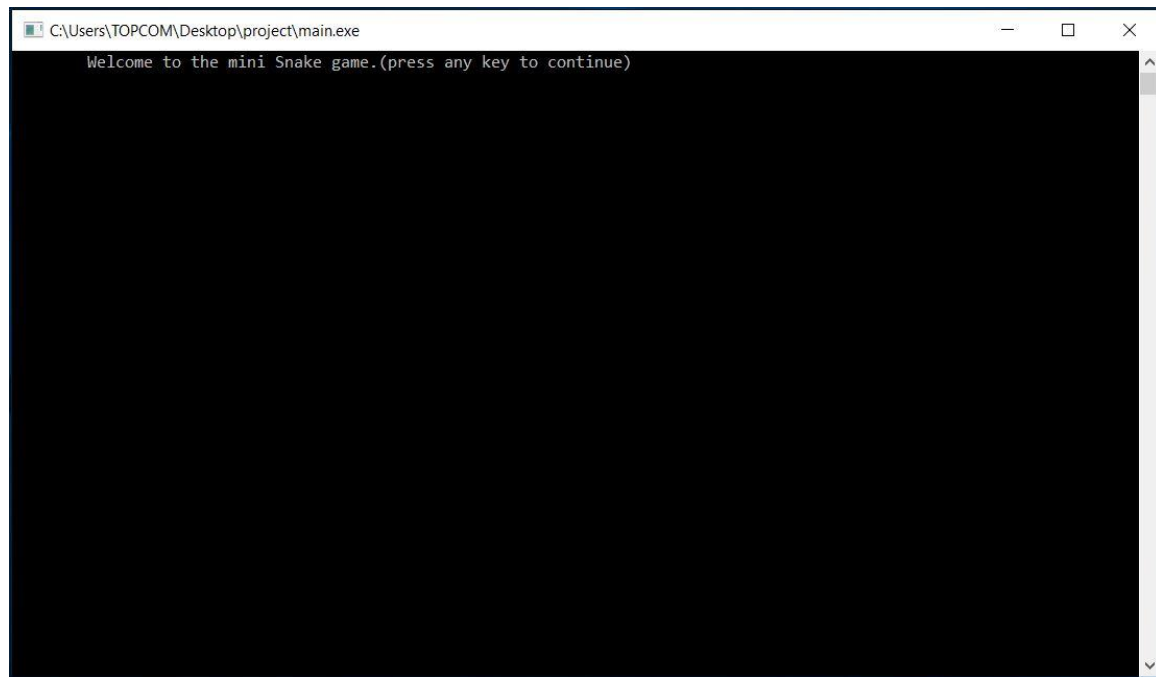
1. หาความรู้เพิ่มเติมจากเว็บไซต์ต่างๆ เพื่อนำมาปรับใช้กับงาน
2. สอบถามผู้ที่มีความรู้ในเรื่องเกี่ยวกับ code เพื่อเป็นที่ปรึกษาในการทำงาน
3. นำความรู้จากในห้องเรียนมาปรับใช้

### ปัญหาและอุปสรรคในการดำเนินงาน:

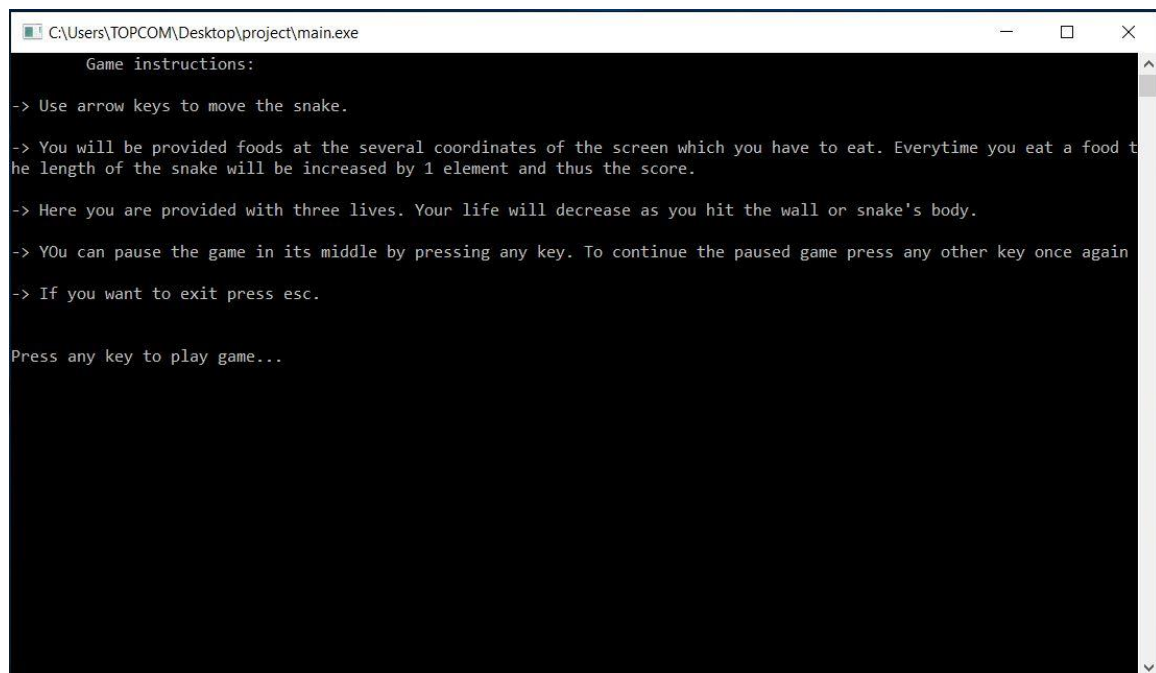
1. หลักการของขั้นตอนวิธี (Algorithm) ยังไม่ค่อยดีจึงทำให้บางครั้งไม่สามารถคิดวิธีการแก้ไขปัญหาคือได้หรือใช้เวลาในการคิดนาน
2. โปรแกรมมีปัญหาเกี่ยวกับรันโปรแกรมเนื่องฟังก์ชันที่นำมาใช้โปรแกรมไม่สนับสนุนจึงเกิดปัญหาตามมา
3. บางชุดคำสั่งที่ใช้ในเกม อาจารย์ไม่ได้สอนในชั้นเรียนจึงจำเป็นต้องทำการศึกษาด้วยตนเอง

# ตัวอย่างการรันของโปรแกรม

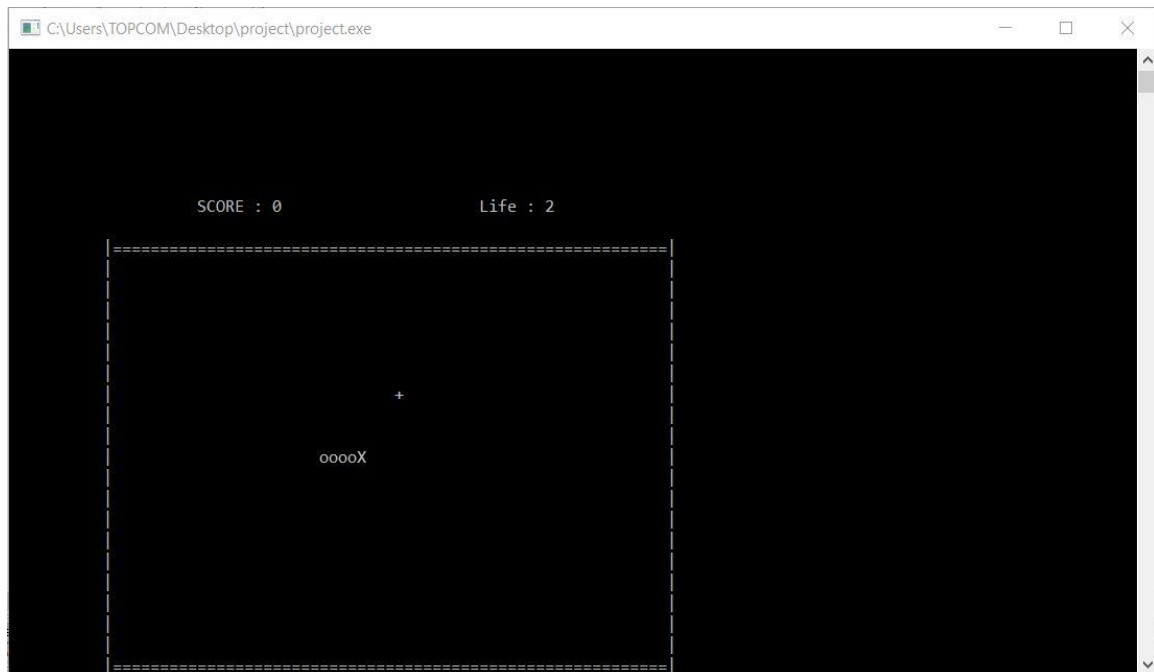
## 1. หน้าแรกเมื่อเข้าโปรแกรม



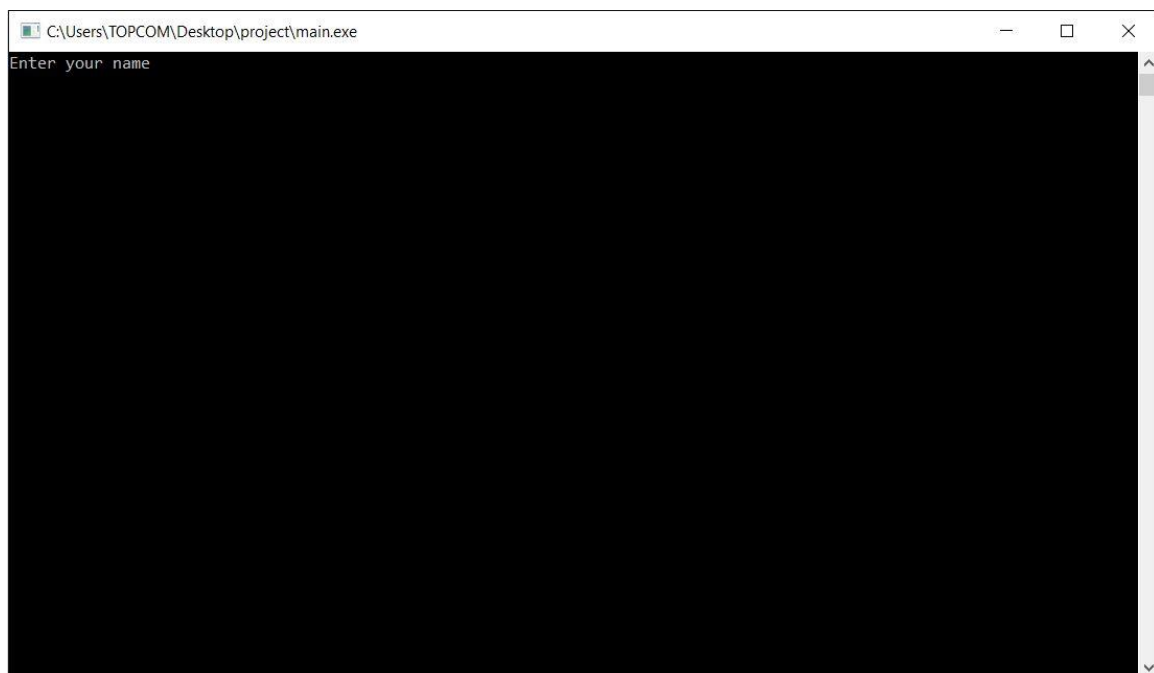
## 2. หน้าที่สองหลังจากกดปุ่ม Enter แสดงรายละเอียดของเกม



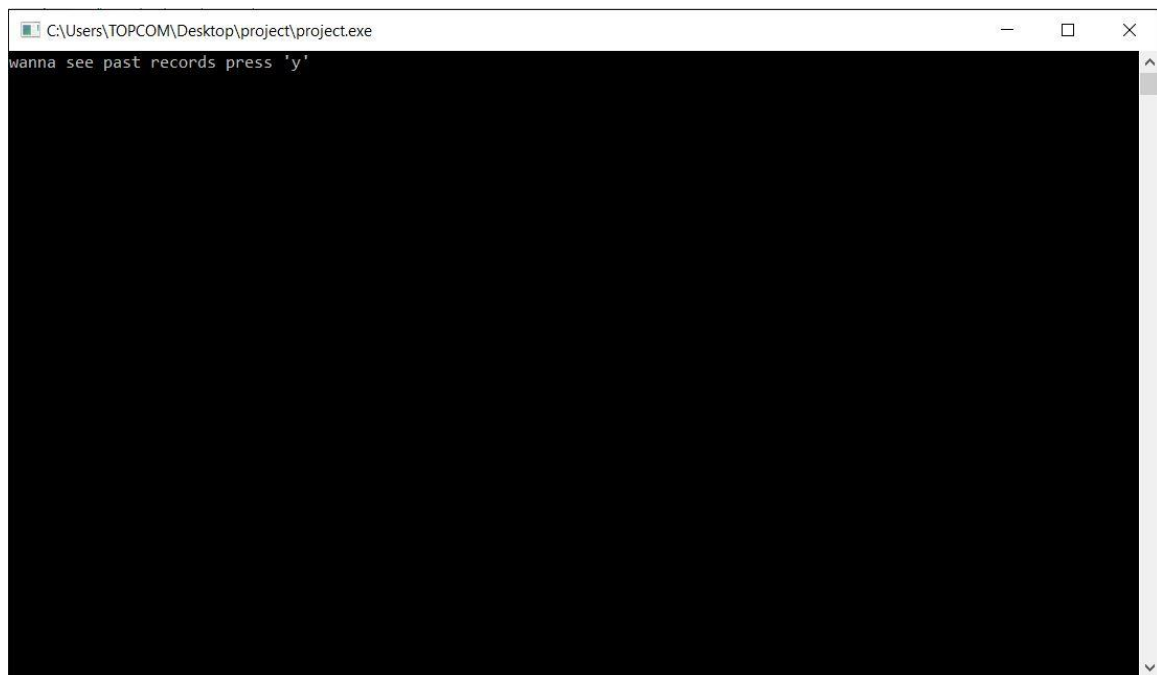
3.หน้าที่สามหลังจากอ่านรายละเอียดของเกมกดปุ่มEnterแล้วก็จะเข้าสู่หน้าจอเกม



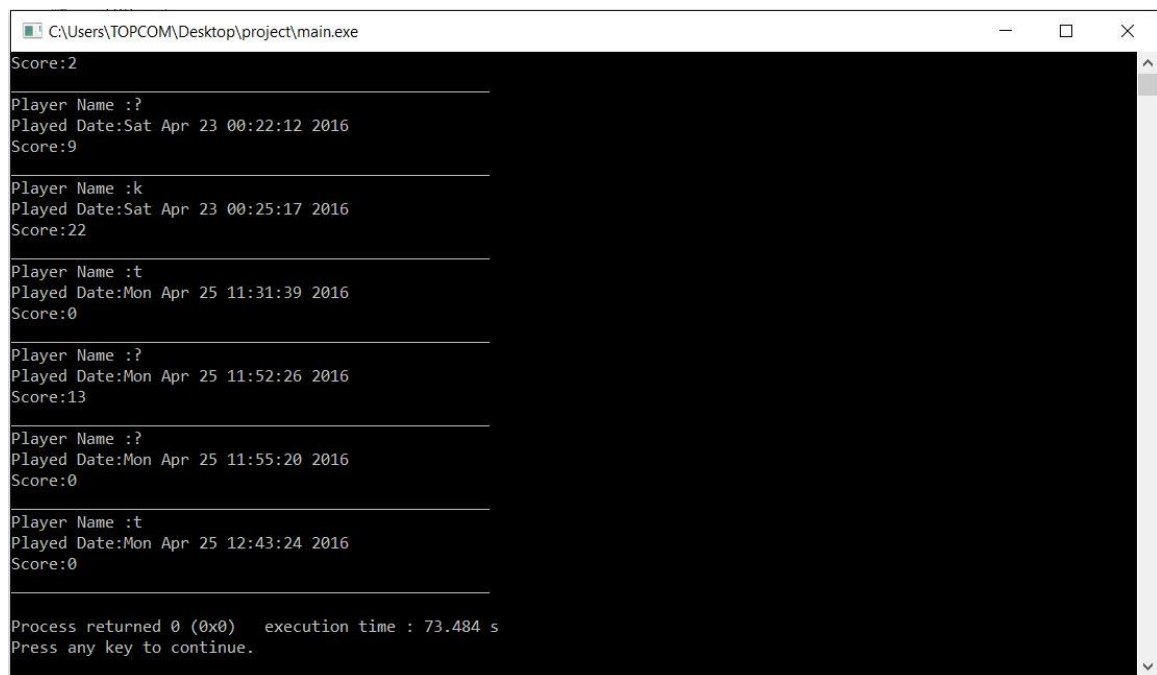
4.หลังจากเล่นเกมจบจะมีให้ใส่ชื่อของผู้เล่น



5.เมื่อใส่ชื่อผู้เล่นเสร็จจะแสดงประวัติการเล่นย้อนหลังหลังจากที่ผู้เล่นกดปุ่ม y



6.หน้าจอแสดงผลสถิติย้อนหลังของผู้เล่น



## เอกสารอ้างอิง

- C Standard Library

สืบค้นจาก [http://www.tutorialspoint.com/c\\_standard\\_library](http://www.tutorialspoint.com/c_standard_library)

- ตาราง ASCII CODE

สืบค้นจาก <http://www.google.co.th/ตารางascii>

- เนื้อหาการเขียน GUI กับภาษาซี

สืบค้นจาก <http://stackoverflow.com/questions/5450047/how-can-i-do-gui-programming-in-c>

- วิธีการใช้ฟังก์ชัน gotoxy

สืบค้นจาก <http://www.programmingsimplified.com/c/conio.h/gotoxy>

- วิธีการใช้ฟังก์ชัน COORD

สืบค้นจาก <http://stackoverflow.com/questions/15770853/how-to-use-setconsolecursorposition-func>