

# 게임프로그래밍

(7\_2\_1 예제 코드 설명)

날짜: 2023.10.04

학번: 2019775045

이름:이상배



```
__modifier_ob__
mirror object to mirror
mirror_object
peration == "MIRROR_X":
mirror_mod.use_x = True
irror_mod.use_y = False
irror_mod.use_z = False
 _operation == "MIRROR_Y"
Irror_mod.use_x = False
lrror_mod.use_y = True
lrror_mod.use_z = False
 _operation == "MIRROR_Z"
 lrror_mod.use_x = False
  __mod.use_y = False
  _rror_mod.use_z = True
 melection at the end -add
   ob.select= 1
   er ob.select=1
   ntext.scene.objects.action
  "Selected" + str(modifice
   irror ob.select = 0
  bpy.context.selected_obje
  Mata.objects[one.name].sel
  int("please select exactle
  OPERATOR CLASSES ----
    vpes.Operator):
    X mirror to the selected
   ject.mirror_mirror_x"
 ontext):
   object is not
```

# 목 차

- 1. 코드 소개 및 출력 결과 7\_2\_1
- 2. 코드에 사용된 함수 설명
  - 1) 헤더파일 및 define 값
  - Intro\_game
  - 3) game\_control
  - 4) print\_number
  - 5) find\_num
  - 6) check\_result
  - 7) main

# 코드 소개 및 출력결과

```
#include <stdio.h>
                                                        31
    #include <time.h>
                                                        32
    #include <conio.h>
                                                        33 □ {
    #include <stdlib.h>
                                                        34
    #define MATRIX_SIZE 5
                                                        35
    #define MAX_NUMBER 5
                                                        36
     #define TIME LIMIT 5
                                                        37
    void intro game(void);
                                                        38
    void game control(void);
                                                        39
    void print number(int n, int cnt[]);
                                                        40
    int find num(int cnt[]);
    void check result(int user, int answer, int cnt[]);
    int main(void)
                                                        42
14
15 □ {
                                                        43
        srand(time(NULL));
16
                                                        44 🖨
17
        intro game();
                                                        45
            game_control();
18
                                                        46
19
        return 0;
                                                        47
20
                                                        48 🖨
21
22
                                                        49
23
     void intro game(void)
                                                        50
24 □ {
                                                        51
25
        printf("정방행렬에 나타난 숫자 중에서 \n");
                                                        52
26
        printf("가장 많이 출현한 숫자를 찾아라!\n\n");
                                                        53
        printf("제 한 시 간 은 5초 입 니 다 .\n\n");
27
                                                        54
28
        printf("아무키나 누르면 시작합니다. \n");
29
        getch();
                                                        55
30 L }
                                                        56
```

```
void game control(void)
   int cnt[MAX NUMBER], answer, user;
   clock t start, end;
   double pst;
   system("cls");
   print number(MATRIX SIZE, cnt);
   answer=find num(cnt);
   printf("\n\n");
   printf("출현횟수가 가장 많은 숫자 입력>");
   start=clock();
   while(!kbhit())
       end=clock();
       pst=(double)(end-start)/CLK TCK;
       if (TIME LIMIT<pst)</pre>
           printf("\n제 한시간을 넘었습니다. \n");
           exit(0);
                                           84
                                           85
                                           86
   user=getch()-'0';
                                           87
   printf("%d", user);
                                           89
   check result(user, answer, cnt);
                                           91
                                           92
                                           93
                                           94
```

95

```
void print number(int n, int cnt[])
   59 □ {
   60
             int i, j, r_num;
   61
             for(i=0;i<MAX_NUMBER;i+=1)</pre>
                  cnt[i]=0;
    62
    63
             for(i=1;i<=n;i+=1)
    64 □
                  for(j=1;j<=n;j+=1)
   66 🖨
    67
                      r_num=rand()%MAX NUMBER;
   68
                      cnt[r num]+=1;
   69
                      printf("%d ", r num);
   70
   71
                  printf("\n");
   72
   73
   74
        int find num(int cnt[])
   76 □ {
             int i. check. max num=0;
    77
             for(i=0;i<MAX NUMBER;i+=1)</pre>
    78
   79 \dot{\Box}
    80
                  if (max_num<=cnt[i])</pre>
    81 🗀
   82
                      max_num=cnt[i];
              check=i:
       return check;
   void check_result(int user, int answer, int cnt[])
90 □ {
       if (user==answer)
           printf("\n맞았습니다. \n");
       else
           printf("\n틀 렸 습 니 다 . \n");
       printf("번호: %d, 횟수 : %d\n", answer, cnt[answer]);
96 L )
```

# 코드 소개 및 출력결과

정방행렬에 나타난 숫자 중에서 가장 많이 출현한 숫자를 찾아라!

제한시간은 5초입니다.

아무키나 누르면 시작합니다.

```
1 2 1 2 3
4 4 3 4 2
2 4 0 0 2
2 3 1 3 3
3 1 4 4 2
출현횟수가 가장 많은 숫자 입력>
```

```
4 2 2 3
 1 2 3 0
 2 2 3 3
 2 2 2 0
출현횟수가 가장 많은 숫자 입력>
제한시간을 넘었습니다.
3 3 3 1 3
 2022
 0 2 1 3
 4 2 4 2
 1 3 3 4
출현횟수가 가장 많은 숫자 입력>3
맞았습니다.
번호: 3, 횟수 : 9
 3 1 0 4
3 0 1 1 1
3 4 4 3 4
2 0 1 1 3
20004
출현횟수가 가장 많은 숫자 입력>4
틀렸습니다.
번호: 1, 횟수 : 7
```

```
1 #include <stdio.h>
2 #include <time.h>
3 #include <conio.h>
4 #include <stdlib.h>
5 #define MATRIX_SIZE 5
6 #define MAX_NUMBER 5
7 #define TIME_LIMIT 5
```

# **함수 설명** (헤더파일 및 define 값)

- stdio.h: 표준 입력/출력 함수를 사용하기 위한 헤더 파일.
- time.h: 시간 관련 함수를 사용하기 위한 헤더 파일.
- conio.h: 일부 플랫폼에서 특정 키보드 입력 함수를 사용하기 위한 헤더 파일.
- stdlib.h: 일반적인 유틸리티 함수를 사용하기 위한 헤더파일.
- MATRIX\_SIZE: 게임 행렬의 크기 (5x5).
- MAX\_NUMBER: 게임에서 사용되는 숫자의 범위 (0부터 MAX\_NUMBER-1까지).
- TIME\_LIMIT: 사용자의 답변 시간 제한 (5초).

### 함수 설명 (Intro\_game)

```
23 void intro_game(void)
24□
{
    printf("정방행렬에 나타난 숫자 중에서\n");
    printf("가장 많이 출현한 숫자를 찾아라!\n\n");
    printf("제한시간은 5초입니다.\n\n");
    printf("아무키나 누르면 시작합니다.\n");
    getch();
30 }
```

- 개임 소개 메시지를 화면에 출력.
- getch(): 사용자가 키를 입력받는 함수.

```
void game control(void)
33 □ {
        int cnt[MAX NUMBER], answer, user;
34
35
        clock t start, end;
36
        double pst;
37
        system("cls");
        print number(MATRIX SIZE, cnt);
39
        answer=find num(cnt);
        printf("\n\n");
        printf("출현횟수가 가장 많은 숫자 입력>");
41
        start=clock();
42
43
        while(!kbhit())
44 E
            end=clock();
            pst=(double)(end-start)/CLK TCK;
47
            if (TIME LIMIT<pst)</pre>
48 E
                printf("\n제한시간을 넘었습니다. \n");
49
50
                exit(0);
51
52
53
        user=getch()-'0';
54
        printf("%d", user);
55
        check result(user, answer, cnt);
```

### 함수 설명 (game\_control)

- cnt 배열을 초기화하고 무작위 숫자를 생성하여 print\_number 함수를 호출하여 화면에 출력.
- find\_num 함수를 호출하여 가장 많이 출현 한 숫자를 찾음.
- 사용자에게 제한 시간 내에 숫자를 입력하 도록 요청하며, 시간을 측정.
- 입력된 숫자와 정답을 비교하고, check\_result 함수를 호출하여 결과를 출력

#### void print number(int n, int cnt[]) 58 59 🗦 { 60 int i, j, r num; 61 for(i=0;i<MAX NUMBER;i+=1)</pre> 62 cnt[i]=0; 63 for(i=1;i<=n;i+=1) 64 🖨 65 for(j=1;j<=n;j+=1) 66 🖨 67 r num=rand()%MAX NUMBER; cnt[r\_num]+=1; 68 printf("%d ", r\_num); 69 70 71 printf("\n"); 72 73

# 함수 설명 (print\_number)

• n 횟수만큼 반복하여 무작위 숫자를 생성하고 cnt 배열에 해당 숫자의 출현 횟수를 누적시킴. 또한, 생성된 숫자를 화면에 출력

#### int find\_num(int cnt[]) 76 ₽ { 77 int i, check, max\_num=0; for(i=0;i<MAX\_NUMBER;i+=1)</pre> 78 **79** 申 if (max\_num<=cnt[i])</pre> 80 **81** $\Rightarrow$ 82 max\_num=cnt[i]; 83 check=i; 84 85 return check; 86 87

# 함수 설명 (find\_num)

• cnt 배열을 검사하여 가장 많이 출현한 숫자와 해당 숫자의 인덱스를 반환.

# 함수 설명 (check\_result)

```
89 void check_result(int user, int answer, int cnt[])
90 {
91     if (user==answer)
92         printf("\n맞았습니다. \n");
93     else
94         printf("\n틀렸습니다. \n");
95     printf("번호: %d, 횟수: %d\n", answer, cnt[answer]);
96 }
```

 사용자가 입력한 답과 정답을 비교하여 정답 여부를 출력. 정답인 경우, "맞았습니다", 오답인 경우, "틀렸습니다"를 출현 횟수와 정답 숫자를 함께 출력.

### **함수 설명** (main)

- srand(time(NULL))를 호출하여 난수 발생기를 초기화.
- intro\_game 함수를 호출하여 게임 소개 메시지를 출력.
- game\_control 함수를 호출하여 게임 플레이를 시작.

# 출처

https://ko.wikipedia.org/wiki/Stdio.h https://ko.wikipedia.org/wiki/Time.h https://ko.wikipedia.org/wiki/Stdlib.h https://en.wikipedia.org/wiki/Conio.h 교수님의 강의자료

