

문제해결기법 2주차 과제

202033762 장민호

- 코드

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
/*
문제해결기법 과제
슬롯머신 프로그래밍
세개의 슬롯, 각 슬롯은 BAR, BELL, LEMON, CHERRY 가 가능하다
1. CHERRY 3개 == JACKPOT
2. CHERRY 1개 == One Dime
3. CHERRY 제외 다른 것 세개 == Nickel
*/
#define BAR 0
#define BELL 1
#define LEMON 2
#define CHERRY 3
#define SLOT_1 0
#define SLOT_2 1
#define SLOT_3 2
#define JACKPOT 0
#define ONE_DIME 1
#define NICKEL 2
#define NOTHING 3
#define SLOT_MAX 3
#define RANDOM_MAX 4
void randomSlot(int machine[])
{
    srand(time(NULL));
    for (int i = 0; i < SLOT_MAX; i++)
    {
        machine[i] = (int)rand() % RANDOM_MAX;
    }
}
// 각 배열에 들어간 숫자가 어떤 데이터를 의미하는지 파악한다.
char *returnEntity(int machine[], int index)
{
    switch (machine[index])
    {
        case BAR:
            return "BAR";
            break;
        case BELL:
            return "BELL";
            break;
        case LEMON:
            return "LEMON";
            break;
        case CHERRY:
            return "CHERRY";
            break;
    }
    return NULL;
}
void printResult(int machine[], int flag)
{

```

```

printf("Welcome to KW Land\n");
printf("Please pull the slot machine !!\n\n");
// 일단 각 슬롯에 들어간 정보를 출력한다.
printf("First is %s\n", returnEntity(machine, SLOT_1));
printf("Second is %s\n", returnEntity(machine, SLOT_2));
printf("Third is %s\n", returnEntity(machine, SLOT_3));
// 대강 10자리 문자열 포인터 생성해준다.
char *msg = malloc(sizeof(char) * 10);
switch (flag)
{
case JACKPOT:
    strcpy(msg, "JACKPOT");
    break;
case ONE_DIME:
    strcpy(msg, "One Dime");
    break;
case NICKEL:
    strcpy(msg, "Nickel");
    break;
default:
    strcpy(msg, "Nothing");
    break;
}
printf("Paid Out : %s\n", msg);
printf("Press any key to continue\n");
}
void searchSlot(int machine[])
{
    int flag = NOTHING;
    // 세 개가 모두 같을 경우
    if (machine[SLOT_1] == machine[SLOT_2] && machine[SLOT_2] == machine[SLOT_3])
    {
        // 1. 모두 CHERRY 인 경우
        if (machine[SLOT_1] == CHERRY)
        {
            flag = JACKPOT;
        }
        // 2. 모두 CHERRY 가 아니고 세개가 모두 같을 경우
        else
        {
            flag = NICKEL;
        }
    }
    // 세 개 중에 하나라도 체리가 있을 경우
    else if (machine[SLOT_1] == CHERRY || machine[SLOT_2] == CHERRY || machine[SLOT_3] ==
CHERRY)
    {
        flag = ONE_DIME;
    }
    printResult(machine, flag);
}
int main()
{
    int slotMachine[SLOT_MAX];
    do
    {
        randomSlot(slotMachine);
        searchSlot(slotMachine);
    } while (getchar()); // Enter을 입력받으면 계속 진행함
    return 0;
}

```

- 실행결과

```
Welcome to KW Land  
Please pull the slot machine !!
```

```
First is  CHERRY  
Second is CHERRY  
Third is  BELL
```

```
Paid Out : One Dime  
Press any key to continue
```

```
Welcome to KW Land  
Please pull the slot machine !!
```

```
First is  BELL  
Second is BELL  
Third is  BELL
```

```
Paid Out : Nickel  
Press any key to continue
```

```
Welcome to KW Land  
Please pull the slot machine !!
```

```
First is  BAR  
Second is CHERRY  
Third is  LEMON
```

```
Paid Out : One Dime
```