

[MSP 2014 풀이 보고서]

Friday, November 21, 2014 4:57 PM

양경석(gaeng4@gmail.com)

MSP 2014 소개.



홈페이지: <http://www.microsoft.com/ko-kr/events/2014/dreamsparkmvaproject/codechallenge/>

공식 카페: <http://cafe.naver.com/mspforever>

[01] (완료)

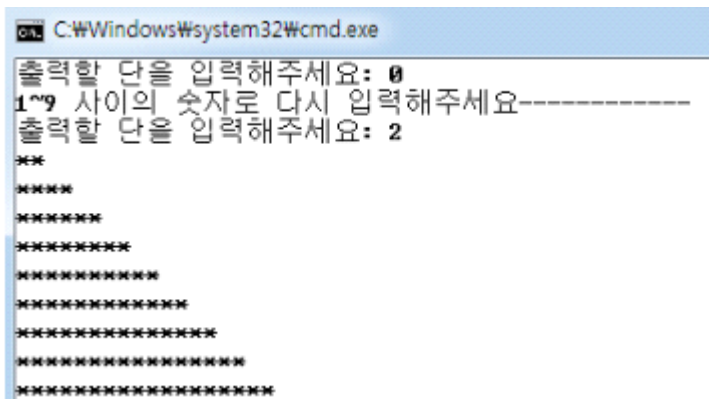
Friday, November 21, 2014 4:55 PM

문제1. 구구단의 단을 입력 받아 *로 출력하라.

입력 받은 수가 1~9 사이의 수인지 확인한다.

잘못 입력한 경우 다시 입력 받는다.

[출력 결과]



소스코드

```
//문제1. 구구단의 단을 입력 받아 *로 출력하라.  
//입력 받은 수가 1~9 사이의 수인지 확인한다.  
//잘못 입력한 경우 다시 입력 받는다.
```

```
#include <stdio.h>
```

```
int main(){  
    int i=0,j=0;  
    int dan=0;
```

```
    while(1){  
        printf("출력할 단을 입력해주세요:");  
        scanf("%d",&dan);  
        if(dan>=1 && dan<=9){  
            printf("1~9입니다.\n");
```

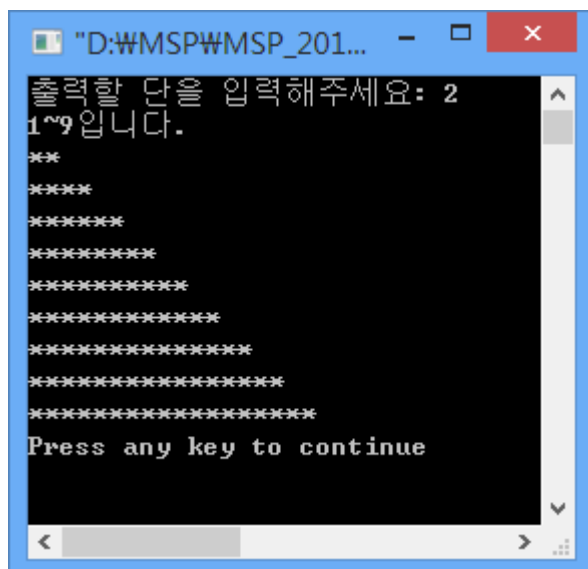
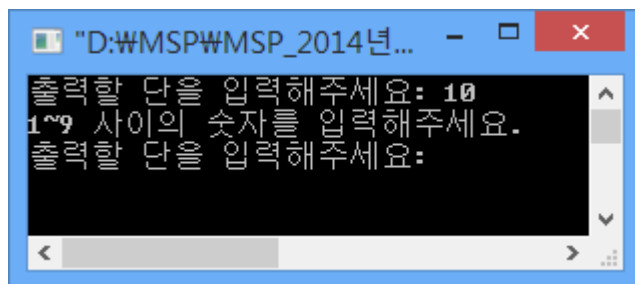
```

    break;
}
else{
    printf("1~9 사이의 숫자를 입력해주세요.\n");
}
}
for(i=1;i<=9;i++){
    //printf("%d x %d = %d\n",dan,i,dan*i);
    for(j=1;j<=(dan*i);j++){
        printf("*");
    }
    printf("\n");
}

return 0;
}

```

결과



[02] (완료)

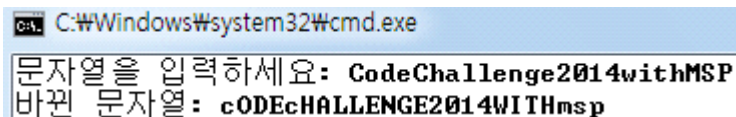
Friday, November 21, 2014 7:18 PM

문제2. 주어진 문장의 대문자를 소문자로, 소문자를 대문자로 변경하는 프로그램을 작성하라.

문장 : CodeChallenge2014withMSP

주어진 문장을 출력 결과와 같이 출력하라.

[출력 결과]



C:\Windows\system32\cmd.exe
문자열을 입력하세요: CodeChallenge2014withMSP
바뀐 문자열: cODEcHALLENGE2014WITHmsp

```
//문제2. 주어진 문장의 대문자를 소문자로,  
//소문자를 대문자로 변경하는 프로그램을 작성하라.
```

```
//문장 : CodeChallenge2014withMSP
```

```
#include <stdio.h>
```

```
int main(){
```

```
char arr[]="CodeChallenge2014withMSP";  
int i=0;
```

```
for(i=0;arr[i]!='\0';i++){  
if(arr[i]>='a' && arr[i]<='z'){  
printf("%c",arr[i]+'A'-'a');  
}//소문자인 경우
```

```
else if(arr[i]>='A' && arr[i]<='Z'){  
printf("%c",arr[i]-'A'+'a');  
}//대문자인 경우
```

```
else if(arr[i]>='0' && arr[i]<='9'){printf("%c",arr[i]);  
} //숫자인 경우-else처리해도 되나, 코드 명확성을 위해  
  
    else{  
    }  
    printf("\n");  
  
    return 0;  
}
```

[03] (완료)

Friday, November 21, 2014 7:39 PM

문제3. 쌍이 맞는 괄호를 판별하는 프로그램을 작성하라.

괄호 문자열을 입력 받는다.

입력 받은 괄호 문자열이 한 쌍의 괄호 기호로 된 "()" 인지 아닌지를 판단해서
그 결과를 YES 와 NO로 나타낸다.

출력 결과와 같이 "(())"라면 "YES", "(()"라면 "NO"이다.

코드 한 줄(80자 이내)을 입력해주세요.
(())()
NO
계속하려면 아무 키나 누르십시오 . . .

코드 한 줄(80자 이내)을 입력해주세요.
((())())()
NO
계속하려면 아무 키나 누르십시오 . . .

코드 한 줄(80자 이내)을 입력해주세요.
(())()(())()
YES
계속하려면 아무 키나 누르십시오 . . .

코드 한 줄(80자 이내)을 입력해주세요.
((())()(())()(())())()
NO
계속하려면 아무 키나 누르십시오 . . .

코드 한 줄(80자 이내)을 입력해주세요.
()()()()()()()()
YES
계속하려면 아무 키나 누르십시오 . . .

코드 한 줄(80자 이내)을 입력해주세요.
()((())())(
NO
계속하려면 아무 키나 누르십시오 . . .

코드 한 줄(80자 이내)을 입력해주세요.
)
NO
계속하려면 아무 키나 누르십시오 . . .

//문제3. 쌍이 맞는 괄호를 판별하는 프로그램을 작성하라.

//괄호 문자열을 입력 받는다.

//입력 받은 괄호 문자열이 한 쌍의 괄호 기호로 된 "(" 인지 아닌지를 판단해서

//그 결과를 YES 와 NO로 나타낸다.

//출력 결과가 "()"라면 "YES", "("라면 "NO"이다.


```

#include <stdio.h>

int main(){
    char arr[1024]={0,};
    int i=0,flag=0,cnt1=0,cnt2=0;

    while(flag!=1){
        printf("괄호를 입력해주세요: ");
        scanf("%s",&arr);

        for(i=0;arr[i]!='\0';i++){
            if(arr[i]!='(' && arr[i]!=''){
                printf("잘못입력하셨습니다. ( or ) 만 입력해주세요.\n");
                flag=0;
                break;
            }
            else{
                flag=1;
            }
        }
        //end for
    } //end while

    for(i=0;arr[i]!='\0' && flag!=0;i++){
        if(arr[i]=='('){ cnt1++; }
        else if(arr[i]==')'){ cnt2++; }
        else {}

        if(cnt1-cnt2<0){printf("NO");break;}
    }
    if( (cnt1-cnt2) == 0 ){printf("YES");}
    else{printf("NO");}

    //printf("cnt1: %d, cnt2: %d\n",cnt1,cnt2);

    return 0;
}

```

cnt1은 (를 의미하는 것으로써, 괄호가 열린 후에 닫혀야 하므로 cnt1이 cnt2보다 항상 크거나 같아야 한다.

모든 괄호들을 다 열고 닫은 경우, cnt1=cnt2가 되어야 하며 이 경우 'YES'가 출력된다.

[04] (완료)

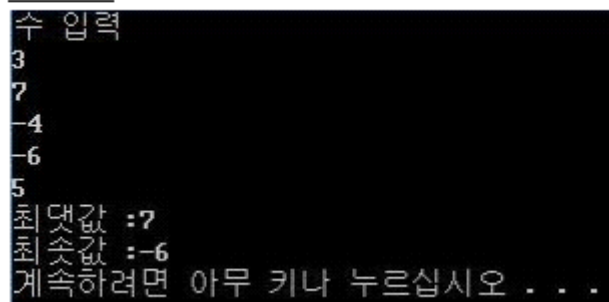
Friday, November 21, 2014 9:16 PM

문제4. 임의의 5개 정수들의 최대값, 최소값을 계산하는 프로그램을 작성하라.

임의의 5개의 정수를 입력 받는다.

입력 받은 정수들 가운데 최대값과 최소값을 찾아 출력하라.

출력결과



```
수 입력
3
7
-4
-6
5
최대값 : 7
최소값 : -6
계속하려면 아무 키나 누르십시오 . . .
```

//문제4. 임의의 5개 정수들의

//최대값, 최소값을 계산하는 프로그램을 작성하라.

//임의의 5개의 정수를 입력 받는다.

//입력 받은 정수들 가운데 최대값과 최소값을 찾아 출력하라.

```
#include <stdio.h>
```

```
int main(){
    char num[5]={0,};
    int temp=0,i=0,j=0,max=0,min=0;

    printf("수 입력: ");
    scanf("%d %d %d %d %d",&num[0],&num[1],&num[2],&num[3],&num[4]);

    max=num[0];
    min=num[0];

    for(i=1;i<5;i++){
        if(min>=num[i]){ min=num[i]; }
        if(max<=num[i]){ max=num[i]; }
    }
    printf("MAX: %d\nMIN: %d\n",max,min);
}
```

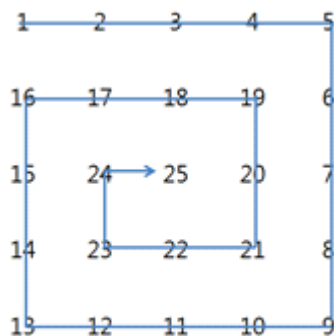
```
    return 0;  
}
```

[05] (완료)

Friday, November 21, 2014 9:30 PM

**문제5. 1부터 20까지 중 임의의 정수를 입력 받아
(정수x정수)를 만족하는 달팽이 배열을 출력하라.**

달팽이 배열이란,
아래 그림과 같이 달팽이 집처럼 돌아 들어가는 배열을 말한다.



아래 그림과 같이 임의의 정수 X값을 입력받고
X행, X열을 만족하는 달팽이 배열을 출력하는 프로그램을 작성하라.

출력결과

```
Please enter any value for nXn array : 10
 1  2  3  4  5  6  7  8  9 10
36 37 38 39 40 41 42 43 44 11
35 64 65 66 67 68 69 70 45 12
34 63 84 85 86 87 88 71 46 13
33 62 83 96 97 98 89 72 47 14
32 61 82 95 100 99 90 73 48 15
31 60 81 94 93 92 91 74 49 16
30 59 80 79 78 77 76 75 50 17
29 58 57 56 55 54 53 52 51 18
28 27 26 25 24 23 22 21 20 19
```

20x20 배열 만들어놓고 해당 번호에 채우고 한번에 출력하기.

```
//문제5. 1부터 20까지 중 임의의 정수를 입력 받아
//(정수x정수)를 만족하는 달팽이 배열을 출력하라.
```

```

#include <stdio.h>

int main(){
    char arr[20][20]={0,};
    int x=0,y=0,i=0,j=0,input=0,num=1,cnt=0,x_sign=1,y_sign=1,flag=1,cnt_x,cnt_y;

    printf("정수를 입력해주세요: ");
    scanf("%d",&input);

    cnt_x=input;
    cnt_y=input-1;

    for(num=1; num<=(input*input); ){

        for(i=0;i<cnt_x;i++,num++){
            x=x+x_sign;
            arr[y][x-1]=num;
        }

        x_sign=-x_sign;
        cnt_x--;

        for(i=0;i<cnt_y;i++,num++){
            y=y+y_sign;
            arr[y][x-1]=num;
        }

        y_sign=-y_sign;
        cnt_y--;
    }

    for(y=0;y<input;y++){
        for(x=0;x<input;x++){
            printf("%3d",arr[y][x]);
        }
        printf("\n");
    }
    return 0;
}

```

[06] (완료)

Monday, November 24, 2014 1:39 AM

문제6. 적립금 프로그램을 만드시오.

힌트

- a. 처음 통장 잔고 금액과 현재 적립금액을 보여준다.
- b. 구매할 물건의 가격을 입력 받는다.
- c. 적립금 사용여부를 묻는다.
 - c-1. 사용 -> 사용할 적립금액 입력.
입력한 값만큼 차감.
 - c-2. 적립 -> 구매금액의 일정금액(5%)을 적립.

- d. 현재 남아있는 적립금의 양을 보여준다.

요구사항

- a. 파일입출력으로 풀어야 하며, 2개의 파일을 필요로 한다.
(통장잔고금액 파일과 적립금액 파일)
* 첨부파일을 확인해주세요.
- b. 예외처리
 - b-1. 적립금 사용시 : 사용할 적립금(입력 값)은 사용 가능한 적립금보다 작거나 같아야 한다.
 - b-2. 구매할 물건의 가격은 통장 잔고보다 같거나 작아야 한다.

유의사항

- a. 자료형은 돈에 관련하여 통상적으로 원 단위로서 계산되기 때문에, int형이라 생각하면 된다.
- b. 이 문제에서는 통장잔고와 적립금을 합쳐 물건을 구매할 수 있다는 가정은 하지 않는다.

***** txt파일명에 대해 확실히 하겠습니다. *****

**일전에 QnA를 통해 파일명을 임의대로 생성하셔도 된다고 답해드렸는데,
앞으로는 파일명에 대해서는 다음과 같이 통일하겠습니다.**

6번 문제 (적립금 문제)

- 적립금.txt
- 통장잔고.txt

이는 검수시 검수용txt 파일명과의 호환 문제때문입니다.

문제 풀이에 혼란을 드려 죄송합니다. 이미 작성해서 제출하신 분들은 상관없으시구요.

앞으로 문제를 푸시는 분들은 이 규칙에 따라 주시기 바랍니다.

감사합니다.

출력결과

```
【 적립금 프로그램 】

... 현재 통장 잔고 : 100000원 , 적립금 : 5000원

구매하실 물건의 가격 : 10000
적립금을 사용하시겠습니까?
--> n
500원 적립되었습니다...

... 현재 통장 잔고 : 90000원 , 적립금 : 5500원
계속하려면 아무 키나 누르십시오 . . .
```

-

【 적립금 프로그램 】

... 현재 통장 잔고 : **90000원**, 적립금 : **5500원**

구매하실 물건의 가격 : **5000**

적립금을 사용하시겠습니까?

--> **y**

사용하실 적립금을 입력하세요 : **5000**

적립금 사용해서서 **0원** 결제되었습니다...

... 현재 통장 잔고 : **90000원**, 적립금 : **500원**

계속하려면 아무 키나 누르십시오 . . .

【 적립금 프로그램 】

... 현재 통장 잔고 : **100000원**, 적립금 : **5000원**

구매하실 물건의 가격 : **200000**

현재 통장 잔고가 부족합니다. 구매하실 물건의 가격을 다시 입력해주세요

구매하실 물건의 가격 : **20000**

적립금을 사용하시겠습니까?

--> **y**

사용하실 적립금을 입력하세요 : **5000**

적립금 사용해서서 **15000원** 결제되었습니다...

... 현재 통장 잔고 : **85000원**, 적립금 : **0원**

계속하려면 아무 키나 누르십시오 . . .

```
#include <stdio.h>
#include <string.h>
#include <math.h>

int main(int argc, char *argv[]){
    FILE * fp1;
    FILE * fp2;
    int jan_go=100000,juk_lip=5000,pay=0,pay_back=0,use_juk_lip=0;
```



```

char ch=0;

fp1=fopen("통장잔고.txt","w+b");
fp2=fopen("적립금.txt","w+b");

fprintf(fp1,"%d",jan_go);
fprintf(fp2,"%d",juk_lip);

while( (jan_go!=0) ){
    printf("현재 통장 잔고는 %d원이고, 적립금은 %d원 입니다.\n",jan_go,juk_lip);
    printf("구매하실 물건의 가격을 입력하세요: ");
    scanf("%d",&pay);
    getchar();

    if( (pay<=jan_go) && (pay>=0) && (jan_go>0) ){

        pay_back=pay*0.05;

        printf("적립금을 사용하시겠습니까? (Y/N) : ");
        scanf("%c",&ch);

        switch(ch){
            case 'y':
            case 'Y':
                while(1){
                    if(juk_lip==0){ printf("적립금이 없습니다.\n"); jan_go=jan_go - pay;
break; }

                    else{
                        printf("현재 적립금은 %d원 있습니다.",juk_lip);
                        printf("사용하실 적립금을 입력하세요: ");
                        scanf("%d",&use_juk_lip);
                        if( (use_juk_lip <= juk_lip) && (use_juk_lip >=0) && ( (juk_lip-
use_juk_lip) >= 0 ) ){
                            jan_go=jan_go - pay + use_juk_lip;
                            juk_lip = juk_lip - use_juk_lip;
                            break;
                        }

                        else{ printf("잘못 입력하셨습니다.\n"); }
                    }//end else
                }//end while

                break;

            case 'n':
            case 'N':
                printf("%d원 적립되었습니다.\n",pay_back);
                juk_lip=juk_lip+pay_back;
                jan_go=jan_go-pay;
                break;
        }//end switch
    }
}

```

```

fclose(fp1);
fclose(fp2);

fp1=fopen("통장잔고.txt","w+b");
fp2=fopen("적립금.txt","w+b");

fprintf(fp1,"%d",jan_go);
fprintf(fp2,"%d",juk_lip);

fclose(fp1);
fclose(fp2);

} //end if

else{ printf("잘못 입력하셨습니다.\n"); } //구매할 물건의 가격(pay)이 부족, 음수입력, 잔
고 부족
} //end while
printf("잔고가 0원입니다. 프로그램을 종료합니다.\n");
return 0;
}

```



적립금



통장잔고

[07] (완료)

Monday, November 24, 2014 1:41 AM

문제7. 겹치는 공강 찾기

힌트 및 요구사항

- a. 파일입출력으로 풀어야 하며, 2개의 시간표 파일이 필요하다.
- b. 시간표 파일(.txt)의 양식은 문제와 함께 제시한 예시 파일을 참고하면 된다.
- c. 출력예시 화면과 같이 2개의 시간표 파일로부터 시간표를 읽어와 출력한 뒤, 겹치는 공강을 출력해주면 된다.

***** txt파일명에 대해 확실히 하겠습니다. *****

일전에 QnA를 통해 파일명을 임의대로 생성하셔도 된다고 답해드렸는데,

앞으로는 파일명에 대해서는 다음과 같이 통일하겠습니다.

7번 문제 (공강 찾기 문제)

- mytime.txt
- friendtime.txt

이는 검수시 검수용txt 파일명과의 호환 문제때문입니다.

문제 풀이에 혼란을 드려 죄송합니다. 이미 작성해서 제출하신 분들은 상관없으시구요.

앞으로 문제를 푸시는 분들은 이 규칙에 따라 주시기 바랍니다.

감사합니다.

출력결과

```

[ 겹치는 공강 찾아주는 프로그램 ]
cf) 공강은 "-"로 수업은 "o"으로 표시됩니다!
< My Timetable >
      MON    TUE    WED    THU    FRI
1      o      o      -      o      o
2      o      -      -      -      -
3      o      o      o      -      o
4      -      o      o      -      -
5      -      -      -      -      o
6      o      -      -      o      o

< Friend's Timetable >
      MON    TUE    WED    THU    FRI
1      o      o      -      o      -
2      o      o      -      o      -
3      o      -      -      -      o
4      -      -      -      o      o
5      -      -      -      -      o
6      o      o      o      o      -

< 겹치는 공강 >
월요일 4교시
월요일 5교시
화요일 5교시
수요일 1교시
수요일 2교시
수요일 5교시
목요일 3교시
목요일 5교시
금요일 2교시
계속하려면 아무 키나 누르십시오 . . .

```

```

#include <stdio.h>
#include <string.h>

int main(int argc, char *argv[])
{
    FILE *fp1;
    FILE *fp2;
    char buf1[1024]={0,};
    char buf2[1024]={0,};
    char mytb[6][5]={0,};
    char frtb[6][5]={0,};

    char weekday[10]="월화수목금";
    int i=0,j=0,k=0;

    fp1=fopen("mytime.txt","r");
    fp2=fopen("friendtime.txt","r");

```

```

while(feof(fp1)==0){ fgets(buf1,sizeof(buf1),fp1); } //파일 내용 불러오기
while(feof(fp2)==0){ fgets(buf2,sizeof(buf2),fp2); } //파일 내용 불러오기

for(i=0,j=0,k=0;i<strlen(buf1);i++){ //내 시간표 2차원 배열에 나눠 담기
    if(buf1[i]=='o' || buf1[i]=='-'){
        mytb[j][k]=buf1[i];

        if(k==4){ k=0; j++; }
        else{ k++; }
    }
}

for(i=0,j=0,k=0;i<strlen(buf2);i++){ //친구 시간표 2차원 배열에 나눠 담기
    if(buf2[i]=='o' || buf2[i]=='-'){
        frtb[j][k]=buf2[i];

        if(k==4){ k=0; j++; }
        else{ k++; }
    }
}

printf("[겹치는 공강 찾아주는 프로그램]\n");
printf("c.f. 공강은 \"-\"로 수업은 \"o\"로 표시됩니다!\n\n");
printf("<My TimeTable>\n");
printf("WtMONWtTUEWtWEDWtTHUWtFRIWn");
for(j=0;j<6;j++){ //내 시간표 출력
    printf("%dWt",j+1);
    for(k=0;k<5;k++){
        printf("%cWt",mytb[j][k]);
    }
    printf("Wn");
}
printf("Wn");

printf("<Friend's TimeTable>\n");
printf("WtMONWtTUEWtWEDWtTHUWtFRIWn");
for(j=0;j<6;j++){ //친구 시간표 출력
    printf("%dWt",j+1);
    for(k=0;k<5;k++){
        printf("%cWt",frtb[j][k]);
    }
    printf("Wn");
}
printf("Wn");

printf("<겹치는 공강>\n");

for(k=0;k<5;k++){
    for(j=0;j<6;j++){
        if( (mytb[j][k]=='-') && (frtb[j][k]=='-') ){ //둘다 공강인 경우 체크
            printf("%c%c요일 %d교시\n",weekday[k*2],weekday[k*2+1],j+1); //한글은 %

```

c%c두개 문자 합쳐서 출력됨.

```
        }  
    }  
}  
  
    return 0;  
}
```



1_7



friendtime



mytime

[08] (완료)

Monday, November 24, 2014 1:41 AM

문제8. Up & Down

힌트 및 요구사항

- a. 난수 생성을 통해 0-50의 수 중 하나가 임의로 생성된다.
- b. 생성된 임의의 수를 아래와 같이 사용자에게 수를 입력받아 맞추면 종료된다.
- c. 시도 횟수(10번)이 초과되면 게임이 종료되도록 하고, 맞출 경우 시도 횟수를 표시한다.
- d. 50을 초과하는 수 또는 음수가 입력되었을 경우 "유효하지 않은 수 입니다."가 출력되도록 하고 시도 횟수에는 포함시키지 않는다.
- e. 입력값을 표시하고, 입력값이 정답보다 큰 수인지 작은 수인지 표시한다.

출력결과

```
0-50의 수 중 하나를 입력하여 정답을 찾아보세요.  
시도 가능 횟수는 10번입니다  
0부터 50 사이의 수를 하나 입력해주세요 > 25  
입력값 : 25 <정답보다 작은 값이 입력되었습니다>  
25부터 50 사이의 수를 하나 입력해주세요 > 40  
입력값 : 40 <정답보다 큰 값이 입력되었습니다>  
25부터 40 사이의 수를 하나 입력해주세요 > 30  
입력값 : 30 <정답보다 작은 값이 입력되었습니다>  
30부터 40 사이의 수를 하나 입력해주세요 > 35  
입력값 : 35 <정답보다 작은 값이 입력되었습니다>  
35부터 40 사이의 수를 하나 입력해주세요 > 36  
입력값 : 36 <정답보다 작은 값이 입력되었습니다>  
36부터 40 사이의 수를 하나 입력해주세요 > 37  
입력값 : 37 정답입니다! (시도 횟수 = 6)  
계속하려면 아무 키나 누르십시오 . . . ■
```

0-50의 수 중 하나를 입력하여 정답을 찾아보세요.
 시도 가능 횟수는 10번입니다

0부터 50 사이의 수를 하나 입력해주세요 > 25
 입력값 : 25 <정답보다 큰 값이 입력되었습니다>

0부터 25 사이의 수를 하나 입력해주세요 > 26
 입력값 : 26 <정답보다 큰 값이 입력되었습니다>

0부터 26 사이의 수를 하나 입력해주세요 > 27
 입력값 : 27 <정답보다 큰 값이 입력되었습니다>

0부터 27 사이의 수를 하나 입력해주세요 > 28
 입력값 : 28 <정답보다 큰 값이 입력되었습니다>

0부터 28 사이의 수를 하나 입력해주세요 > 29
 입력값 : 29 <정답보다 큰 값이 입력되었습니다>

0부터 29 사이의 수를 하나 입력해주세요 > 30
 입력값 : 30 <정답보다 큰 값이 입력되었습니다>

0부터 30 사이의 수를 하나 입력해주세요 > 31
 입력값 : 31 <정답보다 큰 값이 입력되었습니다>

0부터 31 사이의 수를 하나 입력해주세요 > 32
 입력값 : 32 <정답보다 큰 값이 입력되었습니다>

0부터 32 사이의 수를 하나 입력해주세요 > 33
 입력값 : 33 <정답보다 큰 값이 입력되었습니다>

0부터 33 사이의 수를 하나 입력해주세요 > 34
 입력값 : 34 <정답보다 큰 값이 입력되었습니다>

시도 가능 횟수를 초과했습니다.
 계속하려면 아무 키나 누르십시오 . . . ■

```
#include <stdio.h>
#include <time.h>

int main(int argc, char *argv[])
{
    int answer=0,input=0,cnt=1;
    int i=0;
    srand(time(NULL));

    answer=rand()%51;
    printf("answer: %d\n",answer);
    printf("숫자 맞추기 게임\n10번 안에 맞춰야 합니다.\n0~50사이의 수를 입력해주세요!\n");

    while(cnt<=10){
        printf("숫자 입력 (시도횟수 %d/10): ",cnt);
        scanf("%d",&input);

        if(input<0 || input>50){ printf("유효하지 않은 수입니다\n0~50 사이의 수를 다시 입력해주세요\n"); }
        else{
            if(input>answer){printf("down\n");}
            else if(input<answer){printf("up\n");}
            else{printf("정답!\n");break;}
        }
    }
}
```



```
        cnt++;  
    }  
  
}  
printf("Wn정답을 맞추지 못했습니다.\nGAME OVER\n");  
  
return 0;  
}
```

[09] (완료)

Monday, November 24, 2014 1:41 AM

문제9. 2진수 출력.

힌트 및 요구사항

a. 사용자에게 n(2진수의 자릿수)과 k(2진수가 포함해야 할 1의 개수)를 입력 받는다.

예를 들어, n=3 이고 k=1이면 001 010 100

b. n과 k의 범위는 0~10까지 이다.

c. k는 n보다 작거나 같아야 한다.

d. n자리의 2진수 중 1의 개수가 k개인 수를 모두 출력한다.

e. 결과값을 오름차순으로 정렬한다.

출력결과



// 문제9. 2진수 출력.

// a. 사용자에게 n(2진수의 자릿수)과 k(2진수가 포함해야 할 1의 개수)를 입력 받는다.

// 예를 들어, n=3 이고 k=1이면 001 010 100

// b. n과 k의 범위는 0~10까지 이다.

// c. k는 n보다 작거나 같아야 한다.

// d. n자리의 2진수 중 1의 개수가 k개인 수를 모두 출력한다.

// e. 결과값을 오름차순으로 정렬한다.

```
#include <stdio.h>
```

```
#include <math.h>
```

```
char arr[10];
```

```
int n=0,k=0;
```

```
void binary(int i)
```

```
{
```

```
    int a;
```

```
    for(a=0; a < n; a++) {
```

```
        if (i & (1<<a)){
```

```
            arr[a] = 1;
```

```
        }
```

```

        else{ arr[a] = 0; }
    }
}

int main(int argc, char *argv[])
{
    int i=0, j=0, cnt=0, x=0;

    while(1){
        printf("n을 입력해주세요: ");
        scanf("%d",&n);
        if(n>=0 && n<=10){break;}
        else{ printf("0~10 사이의 숫자를 입력해주세요.\n"); }
    }
    while(1){
        printf("k을 입력해주세요: ");
        scanf("%d",&k);
        if(k>=0 && k<=10){break;}
        else{ printf("0~10 사이의 숫자를 입력해주세요.\n"); }
    }

    x=pow(2,n);

    for(i = 0; i < x; i++) {
        binary(i);
        cnt=0;

        for(j=0;j<n; j++) {
            if(arr[j]==1) { cnt++; }
        }

        if(cnt==k){
            for(j=n-1;j>=0;j--){ printf("%d",arr[j]); }
            printf("\n");
        }
    }
    return 0;
}

```

[10] (완료)

Monday, November 24, 2014 1:42 AM

문제10. 노래방 프로그램.

힌트 및 요구사항

- txt 파일에 [번호 / 가수이름 - 노래 제목 / 노래 가사 한 줄]의 데이터를 저장한다.
- 첫 화면에서 [번호 / 가수이름 - 노래 제목] 을 출력한다.
- 사용자가 원하는 노래의 번호를 입력한다.
- 입력 받은 번호에 대한 노래 가사 한 줄을 파일 입출력을 통해 출력한다.

출력결과

* 출력결과에 쓰인 txt파일을 첨부합니다. 확인해주세요.

```
MSP노래방에 오신것을 환영합니다!

번호      가수   -   노래제목
-----
1          sanE&레이나 - 한여름밤의 꿈
2          걸스데이 - 달링
3          태양 - 눈,코,입
4          현아 - 빨개요
5          Bobby - 연결고리

노래 번호를 입력해주세요.
<0을 누르면 노래방프로그램이 종료됩니다.> : 7

**번호가 잘못 입력되었습니다.**

노래 번호를 입력해주세요.
<0을 누르면 노래방프로그램이 종료됩니다.> : 1

무더운 밤 잠은 오지 않고...

노래 번호를 입력해주세요.
<0을 누르면 노래방프로그램이 종료됩니다.> : 0

계속하려면 아무 키나 누르십시오 . . .
```

- //a. txt 파일에 [번호 / 가수이름 - 노래 제목 / 노래 가사 한 줄]의 데이터를 저장한다.
//b. 첫 화면에서 [번호 / 가수이름 - 노래 제목] 을 출력한다.
//c. 사용자가 원하는 노래의 번호를 입력한다.
//d. 입력 받은 번호에 대한 노래 가사 한 줄을 파일 입출력을 통해 출력한다.

```
#include <stdio.h>
```

```

#include <stdlib.h>
int main(int argc, char *argv[])
{
    int i=0,j=0,cnt=0;
    int input=0;

    FILE *fp;
    char str[10][1024]={0,};

    printf("MSP 노래방에 오신 것을 환영합니다!\n\n");
    printf("번호\t가수 - 노래 제목\n");

    fp=fopen("노래방.txt","r");

    for(i=0; feof(fp)==0;i++){
        fgets(str[i],sizeof(str[i]),fp);
    }
    cnt=i;

    fclose(fp);

    for(i=0;i<cnt;i++){
        for(j=0;str[i][j]!='\0';j++){
            printf("%c",str[i][j]);
        }
        printf("\n");
    }

    while(1){
        printf("\n\n(0번을 누르면 노래방 프로그램이 종료됩니다.)\n");
        printf("노래 번호를 입력해주세요 :");
        scanf("%d",&input);

        if(input>=1 && input<=cnt){ //번호에 있는 숫자 누른 경우

            for(i=input-1,j=0;str[i][j]!='\0';j++){ //탭있는곳까지 위치 구함

                for(i=input-1;str[i][j]!='\0';j++){ //탭 있는곳부터 출력
                    printf("%c",str[i][j+1]);
                }
                printf("\n");
            }
        }
        else if(input==0){
            printf("프로그램을 종료합니다.\n");
            exit(0);
        } //0번 누른 경우 프로그램 종료

        else{printf("잘못 누르셨습니다.\n");}
    }
}

```

```
}  
return 0;  
}
```



노래방

[11] (완료)

Monday, November 24, 2014 1:42 AM

문제11. 동전 세는 프로그램

힌트 및 요구사항

- a. 최소한의 동전을 사용하여 거스름돈을 준다.
- b. 음수를 입력하면 절대값을 계산하여 양수로 거스름돈을 계산한다.
 - b-1. 절대값으로 계산될 시 메시지를 출력한다.
- c. 동전의 종류는 다음과 같다.
 - c-1. 4종류의 동전(500원,100원,50원,10원)을 사용한다.
 - c-2. 최소 단위(10원) 이하는 절삭하여 계산하지 않는다.
- d. 입력 금액은 -10억원 이상 ~ 10억원 이하이며 정수(int)로 한정한다.
 - d-1. float, double형으로 입출력하지 않는다.
 - d-2. 동전 개수가 0개여도 출력한다.

출력결과

```
거스름돈 입력
1372
거스름돈<동전 수>
500원 동전: 2개
100원 동전: 3개
50원 동전: 1개
10원 동전: 2개
계속하려면 아무 키나 누르십시오 . . .
```

```
거스름돈 입력
-1372
양수인 1372원으로 계산합니다.
거스름돈<동전 수>
500원 동전: 2개
100원 동전: 3개
50원 동전: 1개
10원 동전: 2개
계속하려면 아무 키나 누르십시오 . . .
```

```
거스름돈 입력
0
거스름돈<동전 수>
500원 동전: 0개
100원 동전: 0개
50원 동전: 0개
10원 동전: 0개
계속하려면 아무 키나 누르십시오 . . .
```

//문제11. 동전 세는 프로그램

```

//힌트 및 요구사항
//a. 최소한의 동전을 사용하여 거스름돈을 준다.
//b. 음수를 입력하면 절대값을 계산하여 양수로 거스름돈을 계산한다.
//  b-1. 절대값으로 계산될 시 메시지를 출력한다.
//c. 동전의 종류는 다음과 같다.
//  c-1. 4종류의 동전(500원,100원,50원,10원)을 사용한다.
//  c-2. 최소 단위(10원) 이하는 절삭하여 계산하지 않는다.
//d. 입력 금액은 -10억원 이상 ~ 10억원 이하이며 정수(int)로 한정한다.
//  d-1. float, double형으로 입출력하지 않는다.
//  d-2. 동전 개수가 0개여도 출력한다.

#include <stdio.h>

int main(int argc, char *argv[])
{
    int input=0;
    int c500=0,c100=0,c50=0,c10=0;

    printf("거스름돈을 입력하세요: ");
    scanf("%d",&input);

    if(input<0){
        input=-input;
        printf("음수를 입력하셨습니다.\n양수인 %d원으로 계산합니다.\n",input);
    }

    c500=input/500;
    input=input-500*c500;
    c100=input/100;
    input=input-100*c100;
    c50=input/50;
    input=input-50*c50;
    c10=input/10;

    printf("거스름돈(동전수)\n");
    printf("500원 동전: %d개\n",c500);
    printf("100원 동전: %d개\n",c100);
    printf(" 50원 동전: %d개\n", c50);
    printf(" 10원 동전: %d개\n", c10);

    return 0;
}

```


[12] (완료)

Monday, November 24, 2014 1:43 AM

문제12. 아나그램(anagram)

요구사항

- 두 문자열에 대하여 서로 '아나그램(anagram)' 관계에 있는지 판별하는 프로그램을 작성한다.
- 아나그램(anagram)이란, 문자열의 문자들이 재배열하여 다른 문자열과 똑같은 문자열을 형성할 수 있으면
두 문자열을 아나그램이라고 한다.
- 예를 들면, 문자열 "THE EYES"와 "THEY SEE"는 서로 아나그램이라고 할 수 있다.
이 두 문자열은 알파벳 E가 3개, H가 1개, T가 1개, S가 1개, Y가 1개로
같은 구성의 알파벳으로 형성되어 있기 때문이다.

요구사항

- 프로그램을 작성할 때, 특수문자나 공백 또는 대소문자 차이는 판별에서 무시하도록 한다.
즉, 어떤 문자가 입력되어도 오직 문자로만 아나그램을 판별하도록 한다.

출력결과

```
Input the 1st sentence: Dormitory
Input the 2nd sentence: Dirty room
Yes! They are anagrams!!
Continue? <y/N>y

Input the 1st sentence: Election results
Input the 2nd sentence: Lie - Let's recount
No! They are not anagrams!!
Continue? <y/N>y

Input the 1st sentence: MOTHER-IN-LAW
Input the 2nd sentence: WOMAN HITLER
Yes! They are anagrams!!
Continue? <y/N>y

Input the 1st sentence: ELEVEN PLUS TWO
Input the 2nd sentence: TWELVE PLUS ONE
Yes! They are anagrams!!
Continue? <y/N>n

Thank you !!
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>
#include <string.h>

int main(int argc, char *argv[])
{
    char sen1[1024]={0,};
    char sen2[1024]={0,};
    char arr1[1024]={0,};
    char arr2[1024]={0,};
    char yesno=0;
```

```

int i=0,j=0,cnt=0;

while(1){
    fflush(stdin);
    printf("input 1st sentence: ");
    gets(sen1);

    printf("input 2nd sentence: ");
    gets(sen2);

    for(i=0,j=0;sen1[i]!='\0';i++){
        if( (sen1[i]>='A' && sen1[i]<='Z') || (sen1[i]>='a' && sen1[i]<='z') )
            {arr1[j]=sen1[i]; j++;}
    }

    for(i=0,j=0;sen2[i]!='\0';i++){
        if( (sen2[i]>='A' && sen2[i]<='Z') || (sen2[i]>='a' && sen2[i]<='z') )
            {arr2[j]=sen2[i]; j++;}
    }

    if( strlen(arr1) != strlen(arr2) ){ printf("no anagram..\Wn"); }
    else{
        for(i=0,cnt=0;i<strlen(arr1);i++){
            for(j=0;j<strlen(arr2);j++){
                if(arr1[i]==arr2[j]){
                    arr1[i]=' ';
                    arr2[j]=' ';
                    cnt++;
                    break;
                }
            }
        }

        if( (cnt==strlen(arr1)) && (cnt!=0)){ printf("YES! ANAGRAM!!\Wn"); }
        else{ printf("no anagram..\Wn"); }
    }

    printf("continue? (y/n) : ");
    scanf("%c",&yesno);

    if(yesno=='n'){ printf("THANK YOU!!\Wn"); break;}
}

return 0;
}

```

공백도 포함해서 받기 위해 gets함수 사용하였음.

일단 문자가 아닌 것은 걸러내고 문자만 새로운 배열에 담음.

배열끼리 비교해서 같은 경우 카운트 증가시키고,

더이상 비교하지 못하도록 문자가 아닌 다른 값(공백문자)으로 채움.

카운트가 0이 아닌 경우(문자가 없는 경우는 아나그램이 아님.)나 cnt가 글자수와 같은 경우에만 아나그램이 됨.

[13] (완료)

Wednesday, December 3, 2014 3:03 AM

문제 13. 입력받은 두 수 사이의 소수 구하기

힌트 및 유의사항

1~300 사이의 정수를 입력 받고, 입력 받은 정수 사이에 소수를 출력한다

예) 1부터 13사이의 소수($1 < \text{소수} < 13$) : 2, 3, 5, 7, 11

출력결과

```
1부터 300사이 정수 두개 입력 : 50 100
53
59
61
67
71
73
79
83
89
97
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>

void sosoo_func(int x){
    int i=0,cnt=0;

    for(i=1;i<=x;i++){
        if(x%i==0){cnt++; }
    }
    if(cnt==2){ printf("%d ",x); }
}

int main(int argc, char *argv[])
{
    int a=0,b=0,i=0;
    printf("1~300 사이의 두 수를 입력하시오:");
```

```
scanf("%d %d",&a,&b);

for(i=a;i<=b;i++){
    sosoo_func(i);
}
printf("Wn");

return 0;
}
```

1부터 그 수까지 모든 수를 모두 나누어 보았을 때, 나누어 떨어지는 것의 갯수가 1과 자기 자신
뿐인 것은 소수이다. 따라서 cnt가 2가 되는 것이 소수이다.

[14] (완료)

Wednesday, December 3, 2014 3:03 AM

문제 14. 숫자를 입력받아 영문으로 변환하기

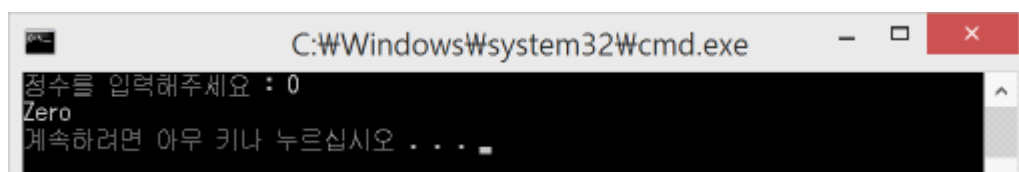
힌트 및 유의사항

- a) 0~9999 까지의 정수를 입력 받고, 입력 받은 정수를 영문으로 변환하여 출력하는 프로그램을 구현하라.
- b) 단, 범위를 벗어난 수(예 - 음수, 10000) 입력 시 Out of Range! 를 출력한다.
- c) 대소문자는 구분하지 않는다.

예시)

0 입력 시 -> 결과값 'zero'
5 입력 시 -> 결과값 'five'
99 입력 시 -> 결과값 'ninety nine'
555 입력 시 -> 결과값 'five hundred fifty five'
3000 입력 시 -> 결과값 'three thousand'
9999 입력 시 -> 결과값 'nine thousand nine hundred ninety nine'
5100 입력 시 -> 결과값 'five thousand one hundred'
-1 입력 시 -> 결과값 'Out of Range!'
10000 입력 시 -> 결과값 'Out of Range!'

출력결과



```
C:\Windows\system32\cmd.exe
정수를 입력해주세요 : 99
결과값: ninety nine 계속하려면 아무 키나 누르십시오 . . .
```

```
C:\Windows\system32\cmd.exe
정수를 입력해주세요 : -1
Out of Range!
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>
```

```
void num_func(int x){
    switch(x){
        case 1: printf(" one");break;
        case 2: printf(" two");break;
        case 3: printf(" three");break;
        case 4: printf(" four");break;
        case 5: printf(" five");break;
        case 6: printf(" six");break;
        case 7: printf(" seven");break;
        case 8: printf(" eight");break;
        case 9: printf(" nine");break;
        default: printf("");
    }//end switch-case
}//end num_func
```

```
void nineteen_under_func(int x){
    switch(x){
        case 1: num_func(x);break;
        case 2: num_func(x);break;
        case 3: num_func(x);break;
        case 4: num_func(x);break;
        case 5: num_func(x);break;
        case 6: num_func(x);break;
        case 7: num_func(x);break;
        case 8: num_func(x);break;
        case 9: num_func(x);break;
        case 10: printf(" ten");break;
        case 11: printf(" eleven");break;
        case 12: printf(" twelve");break;
        case 13: printf(" thirteen");break;
        case 14: printf(" fourteen");break;
        case 15: printf(" fifteen");break;
        case 16: printf(" sixteen");break;
        case 17: printf(" seventeen");break;
        case 18: printf(" eighteen");break;
        case 19: printf(" nineteen");break;
    }
}
```

```
void tenth_func(int x){
```



```

switch(x){
    case 2: printf(" twenty");break;
    case 3: printf(" thirty");break;
    case 4: printf(" fourty");break;
    case 5: printf(" fifty");break;
    case 6: printf(" sixty");break;
    case 7: printf(" seventy");break;
    case 8: printf(" eighty");break;
    case 9: printf(" ninety");break;
}
}
/*
예외 처리할 사항
0~9
10: ten
11: eleven
12: twelve
13: thirteen
14: fourteen
15: fifteen
16: sixteen
17: seventeen
18: eighteen
19: nineteen
20: twenty ~29는 뒤에 one...붙이면 됨
30: thirty
40: fourty
50: fifty
60: sixty
70: seventy
80: eighty
90: ninety
*/

int main(int argc, char *argv[])
{
//    int num=0;
    int a=0,b=0,c=0,d=0;
    int i=0;
//for(i=1000;i<1100;i++){
//    num=i;

//    printf("숫자를 입력하시오");
//    scanf("%d",&num);
    a=num/1000; //천의 자리
    b=(num%1000)/100; //백의자리
    c=(num%100)/10; //십의자리
    d=num%10; //일의자리
//    printf("num: %d, a= %d, b=%d, c=%d, d=%d\n",num,a,b,c,d);

//    printf("[%4d]: ",i);

```

```

if(num==0){ printf("zero"); }
else if( (num>9999) || (num<0) ){ printf("Out of Range!\n"); }
else{
    //1000단위
    if(a!=0){
        num_func(num/1000);
        printf(" thousand");
        num=num%1000;
    }

    //100단위
    if(b!=0){
        num_func(num/100);
        printf(" hundred");
        num=num%100;
    }
    //10단위
    if( (num)>=0 && (num)<=19 ){ nineteen_under_func(num); } //1~19까지

    else if( (num/10)>=2 && (num/10)<=9){ //십의자리 예외 포함시킨것.
        tenth_func(num/10); //10의자리 처리
        num_func(num%10); //1의자리 처리
    }
}
printf("\n");
}
return 0;
}

```

숫자를 세개로 나눔

i) 0 ii) 1~9999 iii) 그 밖의 수

1~9999인 경우

특수의 경우와 일반적인 경우로 나누어서 프로그래밍하였음.

천의자리, 백의자리는 일반적인 경우이나, 십의자리부터 특수 경우가 생겨 케이스 나눠야 함.

1~19 사이의 수, 나머지 수로 나눔.

1~19는 그냥 각각을 써줌.

20, 30 등 뒤가 0으로 끝나는 수는 특이하게 부르는 이름이 있으나, 그 뒤의 일의자리 수는 일반적으로 부르는 대로 부르므로 이 경우에는 앞자리수만 제대로 입력해주면 됨.

num_func, nineteen_under_func, tenth_func 세개의 함수를 사용하여 계속 쓰게되는 코드의 중복을 최소화함.

[15] (완료)

Wednesday, December 3, 2014 3:03 AM

문제 15. 팔찌 만들기

힌트 및 유의사항

- a) 구슬을 이용하여 팔찌를 만들려고 한다. 한 구슬에는 두 가지 색이 들어갈 수 있다.
- b) 색의 종류는 a,b,c, ... , z 까지 26개가 있다고 가정한다. 주머니에서 랜덤으로 구슬을 꺼내어 팔찌를 만든다.
- c) 팔찌를 만들 때 무조건 구슬과 구슬이 접하는 부분의 색은 서로 같도록 구슬을 이어 팔찌를 만든다. 또한 첫 구슬과 마지막 구슬을 마지막에 이을 때도 닿는 부분의 색은 같도록 한다. (아래 그림 참조)
- d) 구슬은 둥글기 때문에 입력한 색 순을 바꿔서 이을 수 있으며, 중복 색도 허락된다. 또한 한 구슬에 같은 색이 두 개 들어간 구슬도 존재한다.
- e) 만약 팔찌 만들기 규칙에 어긋나는 팔찌를 만들면 Fail 메시지를 띄운다.
- f) 팔찌가 만들어지고 나면, 'One More?' 메시지를 띄우고 y/n로 값을 받아 y를 누른 경우 다시 팔찌를 만들도록 한다.
- g) 모든 구슬은 다 사용해야하고, 결과예시는 한가지 경우만 보여주면 된다.



예시)

자신이 생각하는 구슬의 개수(N)를 먼저 입력하고 ($5 < N < 100$), 뽑은 구슬의 색을 하나씩 입력한다.

예1)

구슬의 개수 입력: 5

구슬 색 입력(2개씩 입력)

b c

c d

d a

a b

d b

팔찌 결과: (b c) - (c d) - (d a) - (a b) - (d b)

출력결과

```
Insert the number of bead you're gonna choose(5~100): 5
Insert two colors of each bead <5>
=> 1. : b c
=> 2. : b d
=> 3. : c d
=> 4. : b a
=> 5. : a b

This is YOUR BEADS you picked.
[b, c] [b, d] [c, d] [b, a] [a, b]

It's Possible to make complete Bracelet as follows!!
[a, b]-[b, c]-[c, d]-[d, b]-[b, a]

One MORE?? ;-> : y

=====

Insert the number of bead you're gonna choose(5~100): 5
Insert two colors of each bead <5>
=> 1. : a b
=> 2. : b c
=> 3. : c d
=> 4. : d f
=> 5. : f z

This is YOUR BEADS you picked.
[a, b] [b, c] [c, d] [d, f] [f, z]

It's IMPOSSIBLE to make complete Bracelet!!

One MORE?? ;-> : n
계속하려면 아무 키나 누르십시오 . . .
```

```
#include <stdio.h>
#include <string.h> //for memset
#include <time.h>
#include <stdlib.h>
```

```

int main(){
    int i=0,j=0,number=0; //number: 입력받을 구슬의 갯수
    int input1=0,input2=0;
    char rand_num=0;
    char temp[100][2]={0,};
    char flag[100]={0,};
    char turn[100]={0,}; // 1-2-5-4-3 or 1-5-4-2-3 or ....
    char color_cnt[26]={0,}; //color_cnt[0]= a, [1]=b [2]=c ... 각 숫자의 갯수 셈
    char reverse[26]={0,};
    char swap_temp=0;
    char cnt=0;

    srand(time(NULL));

while(1){
    while(1){
        printf("there are 26(a~z) colors to be chosen\n");
        printf("insert the number of beads (5~100): ");
        scanf("%d",&number);
        if(number>=5 && number<=100){break;}
        else{ printf("wrong number\ninput again\n\n"); }
    }
    memset(flag,1,number);//입력된 갯수만큼 flag를 1로 바꾸어줌.

    //for(i=0;i<100;i++){ printf("flag[%d]: %d\n",i,flag[i]); }

    printf("you chose %d beads\n",number);
    printf("Insert 2 colors each (ex: --> 1. b c )\n");

    for(i=0;i<number;i++){
        printf("--> %d : ",i+1);
        fflush(stdin); //버퍼에 입력된 엔터값을 지움.
        scanf("%c %c",&input1,&input2);
        //printf("%c %c",input1,input2);
        color_cnt[input1-'a']++;
        color_cnt[input2-'a']++;
        temp[i][0]=input1;
        temp[i][1]=input2;
    }//temp[0]~temp[99] 사이에 비즈 구성 넣기 완료.

//비즈가 모양이 만들어지는지 체크
    for(i=0;i<26;i++){
        //printf("%c: %d\n",i+'a',color_cnt[i]);
    }
    for(i=0;i<26;i++){
        if(color_cnt[i]%2==1){
            printf("failed!!\n");
            break;
        }
    }
}

```

```

}
printf("OK!\n"); //failed가 되지 않으면 성공. 비즈 모양 만들어질 환경 OK

//비즈모양 맞추기.

//순번 정하기.
for(i=0;i<number;i++){
    if(i==0){printf("[%c %c]",temp[i][0],temp[i][1]); cnt++;}

    for(j=1;j<number;j++){
        if(temp[i][1]==temp[j][0] && flag[j]==1){//순서대로인 경우
            flag[j]=0;
            printf("-[%c %c]",temp[j][0],temp[j][1]);
            break;
        }

        else if(temp[i][1]==temp[j][1] && flag[j]==1){//순서가 거꾸로인 경우
            flag[j]=0;
            //reverse[i]=1;
            fflush(stdin);
            swap_temp=temp[j][1];
            temp[j][1]=temp[j][0];
            temp[j][0]=swap_temp;
            printf("-#[%c %c]#",temp[j][0],temp[j][1]);
            break;
        }
    }
}

printf("\n");

}
return 0;
}

```

[16] (완료)

Wednesday, December 3, 2014 3:04 AM

문제 16. 볼링 스코어 계산하기

힌트 및 유의사항

- a) 볼링 경기는 총 10개의 핀을 사용한다. 볼링 한 경기는 10개의 프레임으로 이루어진다. 그리고 경기 점수는 매 프레임의 점수를 합산한 것이다.
- b) 마지막 프레임을 제외하면 각 프레임마다 최대 2회까지 투구를 할 수 있는데, 스트라이크인 경우에는 1회만 투구한다.
- c) 첫 회에 스트라이크를 하지 못한 경우에는 한 번 더 투구를 할 기회가 있는데, 두 번째 투구에서 나머지 핀을 모두 쓰러뜨리면 스페어라 하고 그렇지 못한 경우를 미스라고 한다.
- d) 경기는 총 10 프레임으로 진행되고, 경기가 모두 끝나고 나면 각 프레임별 점수를 합산한다.

- 미스: 2회 투구하여 쓰러뜨린 볼링 핀 수가 해당 프레임의 점수가 된다.
 - 스페어: 2회 투구하여 쓰러뜨린 볼링 핀 수(항상 10이 됨)에, 바로 다음에 투구하여 쓰러뜨린 핀 수를 더한 점수가 해당 프레임의 점수가 된다.
 - 스트라이크: 1회 투구하여 쓰러뜨린 볼링 핀 수(역시 항상 10이 됨)에, 바로 다음과 그 다음에 투구하여 쓰러뜨린 핀 수를 더한 점수가 해당 프레임의 점수가 된다.
- 위 규칙에 따르면 스페어와 스트라이크인 경우에는 각각 다음 1회, 그 다음 2회의 투구에서 쓰러뜨린 볼링 핀 개수를 합산해야 하는데, 그렇기 때문에 10번째 프레임에서 총 3회를 던져야 하는 것이다.

3	1	2	3	4	5	6	7	8	9	10	HDP
	52	8 -	62	81	×	61	-7	×	8 /	- -	
0	7	15	23	32	49	56	63	83	93	93	93
	72	71	7 -	8 /	9 -	8 -	7 /	3 /	71	XX -	
	9	17	24	43	52	60	73	90	98	118	118
									211	211	T-4

입력 및 결과값 예시)

-
1. 5 5 8 0 10 7 2 10 10 7 3 9 1 9 1 10 5 5 => 179
 2. 10 10 10 10 10 10 10 10 10 10 10 10 10 => 300
 3. 10 10 10 10 10 10 10 10 10 10 9 0 => 267
 4. 10 10 10 10 10 10 10 10 10 10 0 10 8 => 268
 5. 8 8 10 10 10 10 10 10 10 10 10 10 => error
-

특이점

-스트라이크를 친 후 또 스트라이크를 치면 2회의 투구의 점수를 합산하므로 스트라이크 점수 (10점)+그 다음 투구 점수까지 합산된다.

-파일입출력으로 해결(볼링.txt에서 입력을 읽어서 점수 및 합계 구하기)

-최대 투구수: 9프레임 전부 미스(2x9=18), 10프레임 전부 치는 경우 3

총 21회까지 투구 가능.

-최고점: 300점 (1~9프레임 30점x9+10프레임30점 즉, 전부 스트라이크인 경우)

10프레임 계산방법

1~9프레임과 달리 계산한다.

스트라이크인 경우 2번의 기회가 더 주어지고,

1) 첫 투구가 스트라이크-뒤에 두번 더 투구 가능. 스트라이크는 10점으로 계산

2) 첫 투구가 스트라이크 아닌 경우

i) 두번째 투구가 스페어 혹은 스트라이크인 경우

세번째 투구의 기회가 주어짐. 스페어라도 보너스 점수 없음.

ii) 두번째 투구가 스페어 혹은 스트라이크 아닌 경우

세번째 투구 기회 없음.

```
#include <stdio.h>
```

```
int main(int argc, char *argv[])
{
```

```
    char arr[21]={0,};
```

```
    char arr2[10][4]={0,};    //arr2[][0]: 첫번째 투구, arr2[][1] 두번째 투구, arr2[][2]
```

```
                                //arr2[][3]: 점수 합산, arr[][4]: 보너스점수(스트라이크나 스페
```

어처리시)

```
    char frame10=0; //10프레임에서만 필요한 세번째 투구점수.
```

```
    int i=0,j=0,cnt=0;
```

```
    int sum=0;
```

```
    int error_flag=0; //1인경우 에러임.
```

```
    FILE * fp;
```

```
    fp=fopen("볼링.txt","rt");
```

```
    printf("fp: %dWn",fp);
```

```
    i=0;
```

```
    for(i=0;feof(fp)==0;i++){ //while( fscanf(fp,"%d",&arr[i]) ==1 )
```

```
        fscanf(fp,"%d",&arr[i]);
```

```
//        printf("%dWn",arr[i]);
```

```
    }
```

```
    printf("Wn입력된 내용\n");
```

```
    for(i=0;i<21;i++){
```

```
        printf("%d ",arr[i]);
```

```
    }
```

```
    printf("Wn");
```

```
    for(i=0,j=0;i<21;i++){
```

```
        if(j<9){
```

```
            if(arr[i]==10){ arr2[j][0]=10; arr2[j][1]=0; j++;}
```

```
            else{ arr2[j][0]=arr[i]; i++; arr2[j][1]=arr[i]; j++; }
```

```
        }
```

```
    else if(j==9){    //10프레임의 경우
```

```

        arr2[j][0]=arr[i];
        arr2[j][1]=arr[i+1];
        frame10=arr[i+2]; printf("i: %d j: %d\n",i,j);
        j++;
        break;
    }
    if(j>9){error_flag=1; break; }
}

for(i=0;i<10;i++){//프레임별 점수 합산, 10프레임은 개별합산. 및 에러 체크
    if(i<9){
        arr2[i][2]=arr2[i][0]+arr2[i][1];
        if(arr2[i][2]>10){ error_flag=1; break; }
        else if(arr2[i][0]==10){
            if(arr2[i+1][0]==10){//다음 투구까지 스트라이크인 경우
                arr2[i][3]=arr2[i+1][0]+arr2[i+2][0];
            }
            else if(arr2[i+1][0]!=10){//다음투구는 스트라이크 아닌 경우
                arr2[i][3]=arr2[i+1][0]+arr2[i+1][1];
            }
        } //스트라이크인경우 다음2번투구 점수를 보너스로

        else if(arr2[i][2]==10){ arr2[i][3]=arr2[i+1][0]; } //스페어인 경우 다음 1번
투구 점수를 보너스로
    }
    else{ //10프레임의 경우
        if(arr2[i][0]==10){//첫번째 투구가 스트라이크인 경우
            arr2[i][2]=arr2[i][0]+arr2[i][1]+frame10;
        }
        else if(arr2[i][1]==10){//두번째 투구가 스트라이크인 경우
            arr2[i][2]=arr2[i][0]+arr2[i][1]+frame10;
        }
        else if(arr2[i][0]+arr2[i][1]==10){//두번째 투구가 스페어인 경우
            arr2[i][2]=arr2[i][0]+arr2[i][1]+frame10;
        }
        else{ //세번째 투구의 기회가 없는 경우
            arr2[i][2]=arr2[i][0]+arr2[i][1];
        }
    }
}

if(error_flag==1){ printf("error\n"); }
printf("+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+");
printf("-----+Wn");
printf("|");

for(i=0;i<10;i++){ if(i==9){ printf(" %3d  |",i+1); } else{ printf("%3d  |",i+1);} }
printf("Wn");

```

```
printf("+---+---+---+---+---+---+---+---+---+---+---+---\n");  
  
for(i=0;i<10;i++){  
    if(i<9){  
        if(arr2[i][0]==10){ printf("| x "); }  
        else if(arr2[i][3]==10){ printf("| %2d: /",arr2[i][0]); }  
        else if(arr2[i][0]==0){ printf("| -:%2d",arr2[i][1]); }  
        else if(arr2[i][1]==0){ printf("| %2d: -",arr2[i][0]); }  
        else { printf("| %2d:%2d",arr2[i][0],arr2[i][1]); }  
    }  
  
    else{//i==9 (10프레임의 경우)  
        printf("*%2d:%2d:%2d",arr2[9][0],arr2[9][1],frame10);  
    }  
}  
  
printf(" |");  
printf("\n");  
printf("+---+---+---+---+---+---+---+---+---+---+---+---\n");  
  
for(i=0;i<10;i++){ printf(" |   %2d ",arr2[i][2]); }  
printf(" |");  
printf("\n");  
  
printf("+---+---+---+---+---+---+---+---+---+---+---+---\n");  
--+++\n";  
    for(i=0;i<10;i++){ printf(" |   %2d ",arr2[i][3]); }  
    printf(" |");  
    printf("\n");  
  
printf("+---+---+---+---+---+---+---+---+---+---+---+---\n");  
--+++\n";  
    printf(" |");  
    printf("\n");  
    printf("+---+---+---+---+---+---+---+---+---+---+---+---\n");  
--+++\n";  
sum=0;  
for(i=0;i<10;i++){  
    sum=sum+(arr2[i][2]+arr2[i][3]);  
    printf(" |   %2d ",sum);  
}  
  
printf(" |");  
printf("\n");  
  
return 0;
```


[17] (완료)

Wednesday, December 3, 2014 3:04 AM

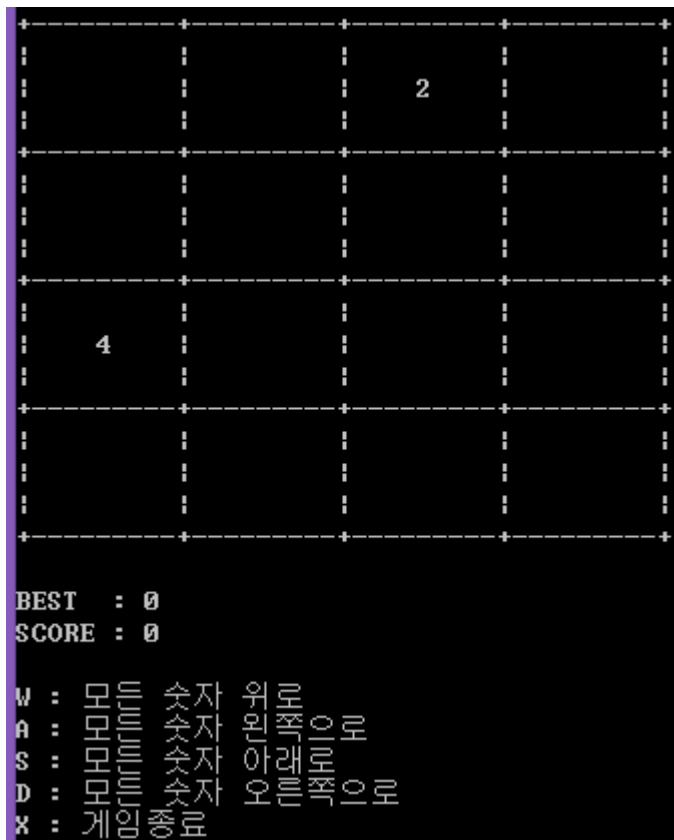
문제 17. 64 게임 만들기

힌트 및 유의사항

- a) 많은 플랫폼으로 구현되어 있는 게임, '2048'의 Easy 버전을 구현한다.
- b) 기본 구조는 '2048'과 완전히 동일하지만 '2048'이 아닌 '64'를 완성하면 된다.
- c) 게임 판은 4*4, 16개의 칸으로 구성되어 있으며 매 판 시작마다 16개의 칸 중에 임의로 두 개의 칸이 선택되어 선택된 칸에 2 또는 4가 랜덤하게 생성된다.
- d) 사용자는 소문자 입력을 포함하여 W,A,S,D 키를 눌러 조작할 수 있다. (4가지 키가 아닌 다른 키를 누를 때 게임의 상태는 변하지 않는다.) 또한 각 키는 따로 엔터(개행문자)를 입력받지 않아도 반응해야 한다. (엔터를 꼭 입력해야 되는 경우 오답처리)
- e) 매 입력시마다 변경되는 게임 판의 상태를 화면에 반영하기 위해 출력 때 `system("cls")`를 이용하여 매 번 화면을 초기화 한 뒤 게임 판을 출력한다. (화면을 지우지 않고 계속 출력하는 경우 오답처리)
- f) 밀어내는 과정에서 병합할 수 있는 숫자가 있다면 모두 병합한다.
* 병합에 대한 자세한 방법은 첨부된 PDF 파일을 참조하세요.
- g) 병합 시에는 점수가 발생한다. 얻는 점수는 병합으로 생성된 숫자의 값이다. 2와 2가 병합되어 4가 생성된 경우 플레이어는 4점을 얻는다.
- h) 키를 누르는 동작으로 인해 숫자의 위치가 변하거나, 병합이 발생했을 경우에만 비어 있는 칸에 2 또는 4의 숫자가 생성된다. 숫자의 위치도 변하지 않고, 병합이 발생하지 않은 경우에는 숫자가 생성되지 않는다.
- i) 숫자 64를 만들어 내면 한 판이 종료된다. 이 점수가 현재까지 진행한 게임의 점수 중 가장 높은 점수인 경우 Best Score에 반영된다.
- j) 플레이 중 모든 칸이 가득차고, 병합될 수 없는 숫자가 없으면 한 판이 종료된다.
- k) 매 판이 종료될 때마다 '게임을 한 번 더 하시겠습니까?'라는 문구를 출력하고 y/n으로 입력을 받는다. 그 외의 입력은 고려하지 않는다.
- l) 게임 진행 중 플레이어가 'X'를 누른 경우에는 프로그램을 종료한다.

출력결과

게임 시작화면입니다.



```
#include <stdio.h>
#include <time.h>

unsigned char x=0,y=0,z=0;
char arr[4][4][2]={0,}; //입력이 없는 경우에 빈칸이 되어야 하므로 문자형으로 입
출력함.
char num[6][2]={{' ','2'},{' ','4'},{' ','8'},{'1','6'},{'3','2'},{'6','4'},};
//2자리의 수를 문자형으로 표현하기 위해 2차원 배열에 각각 담음.

void rand_assign(int _cnt){// 입력 개수만큼 빈칸에 할당하는 함수
    int _i=0;
    for(_i=0;;){ //빈칸에 4나 2 채우기.
        x=rand()%4;
        y=rand()%4;
        z=rand()%2; //rand%2 결과 0 or 1 => 0->'2', 1-> '4'

        if(arr[y][x][1]==0){ //비어있는 곳에 채우기. 숫자가 있는 경우
            arr[y][x][0]=num[z][0]; //뒷자리는 항상 차있으므로
            arr[y][x][1]=num[z][1]; //1의자리인 arr[y][x][1]만 체크하면 됨.
            _i++;
        }
        if(_i==_cnt){ break; } //숫자 2개 배치된 경우 배치 종료.
    }
}

int main(int argc, char *argv[])
{
    char input=0,temp=0,cnt=0;
```

```
int i=0,j=0,k=0,m=0;
srand(time(NULL));
rand_assign(2);
```

while(input!='x'){ //x를 누른 경우 종료됨.

```
printf("random y: %d x: %d z: %d\n",y,x,z);

printf("+-----+-----+-----+-----+Wn");
printf("|      |      |      |      |Wn");//[0][0] [0][1] [0][2] [0][3]
printf("|  %c%c  |", arr[0][0][0],arr[0][0][1]);
printf("  %c%c  |", arr[0][1][0],arr[0][1][1]);
printf("  %c%c  |", arr[0][2][0],arr[0][2][1]);
printf("  %c%c  |Wn",arr[0][3][0],arr[0][3][1]);
printf("|      |      |      |      |Wn");
printf("+-----+-----+-----+-----+Wn");
printf("|      |      |      |      |Wn");//[0][0] [0][1] [0][2] [0][3]
printf("|  %c%c  |", arr[1][0][0],arr[1][0][1]);
printf("  %c%c  |", arr[1][1][0],arr[1][1][1]);
printf("  %c%c  |", arr[1][2][0],arr[1][2][1]);
printf("  %c%c  |Wn",arr[1][3][0],arr[1][3][1]);
printf("|      |      |      |      |Wn");
printf("+-----+-----+-----+-----+Wn");
printf("|      |      |      |      |Wn");//[0][0] [0][1] [0][2] [0][3]
printf("|  %c%c  |", arr[2][0][0],arr[2][0][1]);
printf("  %c%c  |", arr[2][1][0],arr[2][1][1]);
printf("  %c%c  |", arr[2][2][0],arr[2][2][1]);
printf("  %c%c  |Wn",arr[2][3][0],arr[2][3][1]);
printf("|      |      |      |      |Wn");
printf("+-----+-----+-----+-----+Wn");
printf("|      |      |      |      |Wn");//[0][0] [0][1] [0][2] [0][3]
printf("|  %c%c  |", arr[3][0][0],arr[3][0][1]);
printf("  %c%c  |", arr[3][1][0],arr[3][1][1]);
printf("  %c%c  |", arr[3][2][0],arr[3][2][1]);
printf("  %c%c  |Wn",arr[3][3][0],arr[3][3][1]);
printf("|      |      |      |      |Wn");
printf("+-----+-----+-----+-----+Wn");

for(i=0,cnt=0;i<4;i++){ //남은 빈칸 수 체크.
    for(j=0;j<4;j++){
        if(arr[i][j][1]==0){ cnt++; }
    }
}
printf("cnt: %d\n",cnt);

if(cnt==0){ printf("GAME OVER\n"); break;}

input=getch();

printf("%c",input);//입력한 키값 출력(나중에 지우기)
```



```

switch(input){//대소문자 상관없이 w,a,d,s누를 경우에만 기능 구현.
    case 'w':        //위쪽 이동
    case 'W':

        //일단 위로 다 이동시킨 다음에 더하기 체크
        for(j=0;j<4-1;j++){
            for(i=0;i<4;i++){
                for(k=0;k<3;k++){
                    if(arr[j][i][1]==0){//뒷자리만 체크하면 됨.
                        for(m=0;j+m<=2;m++){
                            arr[j+m][i][0]=arr[j+m+1][i][0];arr[j+m][i]
[1]=arr[j+m+1][i][1];
                            arr[j+m+1][i][0]=0;arr[j+m+1][i][1]=0;
                        }
                    }
                }
            }
        }

        for(i=0;i<4;i++){    //2개씩 비교하므로 맨 마지막은 비교하면 안됨.
            for(j=0;j<4-1;j++){
                for(k=0;k<5;k++){ //num[6][] 인 경우는 64(미션성공) 이므로 더할필요
없으므로 체크x
                    if( arr[j][i][0]==num[k][0] && arr[j][i][1]==num[k][1] &&
                        arr[j+1][i][0]==num[k][0] && arr[j+1][i][1]==num[k][1] )
                    {

                        arr[j][i][0]=num[k+1][0]; arr[j][i][1]=num[k+1][1];

                        for(m=0;j+1+m<=2;m++){
                            arr[j+1+m][i][0]=arr[j+1+m+1][i][0];arr[j+1+m][i]
[1]=arr[j+1+m+1][i][1];
                            arr[j+1+m+1][i][0]=0;arr[j+1+m+1][i][1]=0;
                        }
                    }
                }
            }
        }

        rand_assign(1);
        break;

    case 'a':        //왼쪽 이동
    case 'A':

        //일단 전부 왼쪽으로 이동
        for(i=0;i<4-1;i++){
            for(j=0;j<4;j++){
                for(k=0;k<3;k++){
                    if(arr[j][i][1]==0){//뒷자리만 체크하면 됨.

```

```

        for(m=0;i+m<=2;m++){
            arr[j][i+m][0]=arr[j][i+m+1][0];arr[j][i+m][1]=arr[j]
[i+m+1][1];
            arr[j][i+m+1][0]=0;arr[j][i+m+1][1]=0;
        }
        }//4번째 칸에만 차있는 경우때문에 3번 체크해주어야 함.
    }
}

for(i=0;i<4-1;i++){    //2개씩 비교하므로 맨 마지막은 비교하면 안됨.
    for(j=0;j<4;j++){
        for(k=0;k<5;k++){ //num[6][]인 경우는 64(미션성공) 이므로 더할필요
없으므로 체크x
            if( arr[j][i][0]==num[k][0] && arr[j][i][1]==num[k][1] &&
arr[j][i+1][0]==num[k][0] && arr[j][i+1][1]==num[k][1] )
{

                arr[j][i][0]=num[k+1][0]; arr[j][i][1]=num[k+1][1];

                for(m=0;i+1+m<=2;m++){
                    arr[j][i+1+m][0]=arr[j][i+1+m+1][0];arr[j][i+1+m]
[1]=arr[j][i+1+m+1][1];
                    arr[j][i+1+m+1][0]=0;arr[j][i+1+m+1][1]=0;
                }
            }
        }
    }

    rand_assign(1);
    break;

case 'd':        //오른쪽 이동
case 'D':

//일단 전부 오른쪽으로 이동
for(i=3;i>0;i--){
    for(j=0;j<4;j++){
        for(k=0;k<3;k++){
            if(arr[j][i][1]==0){//뒷자리만 체크하면 됨.
                for(m=0;i-m>=1;m++){
                    arr[j][i-m][0]=arr[j][i-m-1][0];arr[j][i-m][1]=arr[j][i-
m-1][1];
                    arr[j][i-m-1][0]=0;arr[j][i-m-1][1]=0;
                }
            }
        }
    }
}
}

```

```

        for(i=3;i>0;i--){ //2개씩 비교하므로 맨 마지막은 비교하면 안됨.
            for(j=0;j<4;j++){
                for(k=0;k<5;k++){ //num[6][] 인 경우는 64(미션성공) 이므로 더할필요
없으므로 체크x
                    if( arr[j][i][0]==num[k][0] && arr[j][i][1]==num[k][1] &&
                        arr[j][i-1][0]==num[k][0] && arr[j][i-1][1]==num[k][1] )
{
                        arr[j][i][0]=num[k+1][0]; arr[j][i][1]=num[k+1][1];

                        for(m=0;i-1-m>=1;m++){
                            arr[j][i-1-m][0]=arr[j][i-1-m-1][0];arr[j][i-1-m]
[1]=arr[j][i-1-m-1][1];
                            arr[j][i-1-m-1][0]=0;arr[j][i-1-m-1][1]=0;
                        }
                    }
                }
            }

            rand_assign(1);
            break;

        case 's': //아래쪽 이동
        case 'S':

            //일단 전부 아래쪽으로 이동
            for(i=0;i<4;i++){
                for(j=3;j>0;j--){
                    for(k=0;k<3;k++){
                        if(arr[j][i][1]==0){//뒷자리만 체크하면 됨.
                            for(m=0;j-m>=1;m++){
                                arr[j-m][i][0]=arr[j-m-1][i][0];arr[j-m][i][1]=arr[j-
m-1][i][1];
                                arr[j-m-1][i][0]=0;arr[j-m-1][i][1]=0;
                            }
                        }//4번째 칸에만 차있는 경우때문에 3번 체크해주어야 함.
                    }
                }
            }

            for(i=0;i<4;i++){ //2개씩 비교하므로 맨 마지막은 비교하면 안됨.
                for(j=3;j>0;j--){
                    for(k=0;k<5;k++){ //num[6][] 인 경우는 64(미션성공) 이므로 더할필요
없으므로 체크x
                        if( arr[j][i][0]==num[k][0] && arr[j][i][1]==num[k][1] &&
                            arr[j-1][i][0]==num[k][0] && arr[j-1][i][1]==num[k][1] )
{

```

```

        arr[j][i][0]=num[k+1][0]; arr[j][i][1]=num[k+1][1];

        for(m=0;j-1-m>=1;m++){
            arr[j-1-m][i][0]=arr[j-1-m-1][i][0];arr[j-1-m][i]
[1]=arr[j-1-m-1][i][1];
            arr[j-1-m-1][i][0]=0;arr[j-1-m-1][i][1]=0;
        }
    }
}

    rand_assign(1);
    break;

default:
    break;
}

system("cls");

}
    return 0;
}

```