

〈로봇 보고서〉

3일차_최우진

1.

〈메인 함수〉

```
int main(int argc, const char * argv[]) {
    int ss[5][5]={0};
    int n=1;
    //위쪽
    for (int i=0; i<2; i++) {
        for (int j=0+i; j<5-i; j++) {
            ss[i][j]+=n;
            n++;
        }
    }
    //중간
    for (int i=2; i<3; i++) {
        for (int j=2; j<3; j++) {
            ss[i][j]+=n;
            n++;
        }
    }
    //아래쪽
    for (int i=3; i<5; i++) {
        for (int j=1-(i-3); j<4+(i-3); j++) {
            ss[i][j]+=n;
            n++;
        }
    }
    //출력
    for (int i=0; i<5; i++) {
        for (int j=0; j<5; j++) {
            printf("%3d",ss[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

->3파트로 나눴습니다. 위,중간,아래 비어있는 부분은 0이 출력되게 구현했습니다.

출력 결과

```
1  2  3  4  5
0  6  7  8  0
0  0  9  0  0
0 10 11 12  0
13 14 15 16 17
Program ended with exit code: 0
```

2.

<메인 함수>

```
int main(int argc, const char * argv[]) {
    int n1,n2,size=0;
    printf("N1 : ");
    scanf("%d",&n1);
    printf("N2 : ");
    scanf("%d",&n2);

    //이중포인터로 배열안에 입력 받은 값 넣기
    int **Arr = (int **)malloc (sizeof(int*)*n1);
    for (int i=0;i<n1;i++){
        Arr[i] = (int *)malloc (sizeof(int)*n2);
    }
    //입력
    int num=1;
    for (int i=0; i<n1; i++) {
        for (int j=0; j<n2; j++) {
            Arr[i][j]=num++;
        }
    }
    //출력
    for (int i=0; i<n2; i++) {
        for (int j=0; j<n1; j++) {
            printf("%d ",Arr[j][i]);
        }
        printf("\n");
    }

    return 0;
}
```

->동적할당을 이용하여 배열안에 변수를 넣었습니다. 그 후 입력받게 하고 출력할때 행열을 반대로 해서 원하는 값이 나오도록 구현했습니다.

출력 결과

```
N1 : 3
N2 : 4

1 2 3 4 5 6 7 8 9 10 11 12
1 5 9
2 6 10
3 7 11
4 8 12
Program ended with exit code: 0
```

3.

<메인 함수>

```
int main(int argc, const char * argv[]) {
    int n1, n2;
    int x = -1, y = 0, k = 0;
    printf("n1 n2 : ");
    scanf("%d %d", &n1, &n2);
    //동적할당으로 배열 만들기
    int **tmp = (int **)malloc (sizeof(int*)*n1);
    for (int i=0; i<n1; i++){
        tmp[i] = (int *)malloc (sizeof(int)*n2);
    }
    //밑에서 조건 걸기 위해 모두 0으로 초기화
    for (int i=0; i<n1; i++) {
        for (int j=0; j<n2; j++) {
            tmp[i][j]=0;
        }
    }

    while (k != (n1*n2)) { //두개 곱한게 k라면 다 채운거 니깐 할출하게 조건 검
        //오른쪽
        while (((x+1)<n2) && (tmp[y][x + 1] == 0)) {
            tmp[y][++x] = ++k;
        }
        //아래
        while (y + 1 < n1 && tmp[y + 1][x] == 0) {
            tmp[++y][x] = ++k;
        }
        //왼쪽
        while (x - 1 >= 0 && tmp[y][x - 1] == 0) {
            tmp[y][--x] = ++k;
        }

        //위
        while (y - 1 >= 0 && tmp[y - 1][x] == 0) {
            tmp[--y][x] = ++k;
        }
    }
    //출력
    for (int i = 0; i < n1; i++) {
        for (int j = 0; j < n2; j++) {
            printf(" %2d", tmp[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

-> 동적할당 이용해서 배열 선언하고 일단 모두 0으로 초기화 했습니다. (뒤에 알고리즘 조건에 걸때 사용하기 위해) 알고리즘은 우선 k라는 값을 한 번 이동할 때마다 증가하게 설정하면서 그게 $n1*n2$ 값과 같아지면 코드 그만 돌아가게 설정했습니다.(그 만큼만 돌리면 되니깐) 그리고 오른쪽 아래쪽 왼쪽 위쪽으로 가게 했는데 기본 알고리즘은 다 똑같이 했습니다. 이동시킬때 다음 배열안에 0이 안들어 있거나 입력받은 $n1, n2$ 값 보다 멀리가면 멈추도록 구현했습니다.

출력 결과

```
n1 n2 : 4 3
  1  2  3
10 11  4
 9 12  5
 8  7  6
Program ended with exit code: 0
```

4.

```
int main(int argc, const char * argv[]) {
    int N,tmp,sum=0;
    float avg;
    printf("몇 개의 원소를 할당하겠습니까? : ");
    scanf("%d",&N);

    //동적할당 이용해서 배열에 포인터 넣기
    int *Arr = (int*)malloc(sizeof(int)*N);
    for (int i=0; i<N; i++) {
        printf("정수형 데이터 입력:");
        scanf("%d",&Arr[i]);
        sum+=Arr[i];
    }

    //정렬
    for (int i=0; i<N-1; i++) {
        for (int j=1; j<N; j++) {
            if(Arr[i]>Arr[j]){
                tmp=Arr[i];
                Arr[i]=Arr[j];
                Arr[j]=tmp;
            }
        }
    }
    //평균
    avg=sum/N;

    //출력
    printf("최대값: %d\n",Arr[N-1]);
    printf("최소값: %d\n",Arr[0]);
    printf("전체합: %d\n",sum);
    printf("평균: %f\n",avg);

    return 0;
}
```

->동적할당을 이용해서 배열안에 변수를 넣었습니다. 그 후 버블정렬 사용하여 정렬시키고 첫 배열과 마지막 배열을 각각 최소값, 최대값으로 설정했습니다. 전체합과 평균을 구하기 위해 sum 변수를 입력 받을때마다 더해주게 해서 전체합을 sum으로 표현했습니다. 평균은 N만큼 나눈값으로 구현했습니다.

출력 결과

```
몇 개의 원소를 할당하겠습니까? : 7
정수형 데이터 입력:1
정수형 데이터 입력:2
정수형 데이터 입력:3
정수형 데이터 입력:4
정수형 데이터 입력:5
정수형 데이터 입력:6
정수형 데이터 입력:7
최대값: 7
최소값: 1
전체합: 28
평균: 4.000000
Program ended with exit code: 0
```

5.

<메인 함수>

```
int main(int argc, const char * argv[]) {
    int n,k,num=1;
    printf("N : ");
    scanf("%d", &n);

    //동적할당으로 배열에 입력 받은 변수 넣기
    int **arr = (int **)malloc (sizeof(int*)*n);
    for (int i=0;i<n;i++){
        arr[i] = (int *)malloc (sizeof(int)*n);
    }

    //알고리즘 t=i, row=j, col=k
    for(int i=0; i<n*2; i++){
        for(int j=0; j<n; j++){
            k = i-j;
            //할당한거 밖으로 안뛰어나가게 하기 위해 설정
            if((0<=k) && (k<n))
                arr[j][k] = num++;
        }
    }

    //출력
    for(int i=0; i<n; i++){
        for(int j=0; j<n; j++){
            printf("%2d ", arr[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

-> 동적할당써서 배열에 변수 넣어서 사용했습니다. 알고리즘은

```
0 0  1 -1  2 -2  3 -3  4 -4
0 1  1  0  2 -1  3 -2  4 -3
0 2  1  1  2  0  3 -1  4 -2
0 3  1  2  2  1  3  0  4 -1
0 4  1  3  2  2  3  1  4  0
0 5  1  4  2  3  3  2  4  1
0 6  1  5  2  4  3  3  4  2
0 7  1  6  2  5  3  4  4  3
0 8  1  7  2  6  3  5  4  4
0 9  1  8  2  7  3  6  4  5
```

이런식으로 for문 돌아가게 설정한 후 조건에 양수 일때만 num++하면서 저장하게 설정하여 구현했습니다.

출력결과

```
N : 5
1  2  4  7 11
3  5  8 12 16
6  9 13 17 20
10 14 18 21 23
15 19 22 24 25
Program ended with exit code: 0
```