

## <로봇 보고서>

2일차\_최우진

1.

### <메인 함수>

```
int main(int argc, const char * argv[]) {
    int size = 0, type = 0;
    printf("사이즈와 종류를 입력하시오.");
    scanf("%d %d", &size, &type);
    if(type == 1)
    {
        type1(size);
    }
    else if(type == 2)
    {
        type2(size);
    }
    else if(type == 3)
    {
        type3(size);
    }
    else if(type == 4){
        type4(size);
    }
    else{
        printf("1~4까지의 숫자만 입력하세요.");
        return 0;
    }
    return 0;
}
```

-> 변수를 선언하고 입력 받은 후 선택한 type에 따라 함수를 호출하게 구현했습니다.

### <type1 함수>

```
void type1(int size){
    for(int i = 0; i < size/2; i++){
        for(int j = 0; j < i+1; j++){
            printf("*");
        }
        printf("\n");
    }
    if(size%2 == 1){
        for(int i = 0; i < size/2+1; i++){
            printf("*");
        }
        printf("\n");
    }
    for(int i = size/2; i > 0; i--){
        for(int j = 0; j < i; j++){
            printf("*");
        }
        printf("\n");
    }
}
```

-> 파트를 나눠서 구현했습니다. 짝수일때와 홀수일때 가운데 \* 있냐 없냐 차이를 가지고 3파트로 나눠서 구현했습니다.

출력 결과

사이즈와 종류를 입력하시오. 5 1

```
*
**
***
**
*
Program ended with exit code: 0
```

<type2 함수>

```
void type2(int size) {
    if(size%2 == 1){
        int cnt = 1;
        for(int j = 0; j < size/2; j++){
            for(int k = 0; k < size/2+1-cnt; k++){
                printf(" ");
            }
            for(int k = 0; k<cnt; k++){
                printf("a");
            }
            cnt++;
            printf("\n");
        }

        for (int i = 0; i < size / 2 + 1; i++) {
            printf("a");
        }
        printf("\n");

        int cnt2 = size/2;
        for(int j = 0; j<size/2; j++){
            for(int k = 0; k < size/2+1-cnt2; k++){
                printf(" ");
            }
            for(int k = 0; k<cnt2; k++){
                printf("a");
            }
            cnt2--;
            printf("\n");
        }
    }
    else{
        int cnt = 1;
        for(int j = 0; j < size/2; j++){
            for(int k = 0; k < size/2-cnt; k++){
                printf(" ");
            }
            for(int k = 0; k<cnt; k++){
                printf("a");
            }
            cnt++;
            printf("\n");
        }

        int cnt2 = size/2;
        for(int j = 0; j<size/2; j++){
            for(int k = 0; k < size/2-cnt2; k++){
                printf(" ");
            }
            for(int k = 0; k<cnt2; k++){
                printf("a");
            }
            cnt2--;
            printf("\n");
        }
    }
}
```

->type2는 짝수일때와 홀수를 먼저 나누고 출력하게 구현했습니다.

출력결과

사이즈와 종류를 입력하시오.5 2

```
*
**
***
**
*
Program ended with exit code: 0
```

<type3 함수>

```

void type3(int size){
    int halfSize = (size + 1) / 2;

    // 위쪽
    for (int i = 1; i <= halfSize; i++) {
        for (int j = 1; j <= size; j++) {
            if (j >= i && j <= size - i + 1) {
                printf("*");
            } else {
                printf(" ");
            }
        }
        printf("\n");
    }

    // 아래쪽
    for (int i = halfSize - 1; i >= 1; i--) {
        for (int j = 1; j <= size; j++) {
            if (j >= i && j <= size - i + 1) {
                printf("*");
            } else {
                printf(" ");
            }
        }
        printf("\n");
    }
}

```

->역시 위쪽과 아래쪽으로 나눠서 구현했습니다.

출력결과

```

사이즈와 종류를 입력하시오.5 3
*****
***
*
***
*****
Program ended with exit code: 0

```

<type4 함수>

```

void type(int size){
    if(size%2 == 1){
        int cnt = size/2+1;
        for(int j = 0; j < size/2; j++){
            for(int k = 0; k<size/2+1-cnt; k++){
                printf(" ");
            }
            for(int k = 0; k < cnt; k++){
                printf("*");
            }
            cnt--;
            printf("\n");
        }

        for(int j = 0; j<size/2; j++){
            printf(" ");
        }
        printf("\n");

        int cnt2 = 2;
        for(int j = 0; j < size/2; j++){
            for(int k = 0; k<size/2; k++){
                printf(" ");
            }
            for(int k = 0; k < cnt2; k++){
                printf("*");
            }
            cnt2++;
            printf("\n");
        }
    }
    else{
        int cnt = size/2+1;
        for(int j = 0; j < size/2; j++){
            for(int k = 0; k<size/2+1-cnt; k++){
                printf(" ");
            }
            for(int k = 0; k < cnt; k++){
                printf("*");
            }
            cnt--;
            printf("\n");
        }

        int cnt2 = 2;
        for(int j = 0; j < size/2; j++){
            for(int k = 0; k<size/2; k++){
                printf(" ");
            }
            for(int k = 0; k < cnt2; k++){
                printf("*");
            }
            cnt2++;
            printf("\n");
        }
    }
}

```

->역시 홀수일때와 짝수일때로 구분해서 구현했습니다.

출력 결과

```

사이즈와 종류를 입력하시오.5 4
***
**
*
**
***
Program ended with exit code: 0

```

2.

<메인 함수>

```

int main(int argc, const char * argv[]) {
    int array[8] = {0};
    for(int i = 0; i < 8; i++){
        printf("%d : ",i);
        scanf("%d",&array[i]);
    }
    for(int j = 0; j < 8; j++){
        printf("%d ",array[j]);
    }
    printf("\n");
    selectionSort(array);
}

```

->메인함수에서 배열 입력받고 함수선언했습니다.

### <선택 정렬 함수>

```

//선택정렬 알고리즘
void selectionSort(int arr[]) {
    for (int i = 7; i > 0; i--) {
        int max = i;
        for (int j = i - 1; j >= 0; j--) {
            if (arr[j] > arr[max]) {
                max = j;
            }
        }
        if (max != i) {
            swap(&arr[i], &arr[max]);
        }
        for(int j = 0; j < 8; j++){
            printf("%d ",arr[j]);
        }
        printf("\n");

        //다 변환 했으면 바꾸는 조건문
        if(arr[0]<arr[1] && arr[1]<arr[2] && arr[2]<arr[3] && arr[3]<arr[4] &&
            arr[4]<arr[5] && arr[6]<arr[7] ){
            break;
        }
    }
}

```

->선택 정렬 알고리즘으로 함수화했습니다. sorting되는 과정 보이게 구현하면서 완벽히 sorting되면 멈추도록 구현했습니다.

### <값 변환 함수>

```

//값 변환해주는 함수
void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

```

-> 한 값잡고 큰지 안큰지 비교한후 교환하는 과정이 선택 정렬 알고리즘에 필

요한데 이 부분을 함수로 만들어서 처리했습니다.

출력 결과

```
0 : 14
1 : 2
2 : 10
3 : 5
4 : 1
5 : 3
6 : 17
7 : 7
14 2 10 5 1 3 17 7
14 2 10 5 1 3 7 17
7 2 10 5 1 3 14 17
7 2 3 5 1 10 14 17
1 2 3 5 7 10 14 17
0 : 1
1 : 2
2 : 3
3 : 5
4 : 7
5 : 10
6 : 14
7 : 17
Program ended with exit code: 0
```

3.

<메인 함수>

```
int main(int argc, const char * argv[]) {
    char ss[10];
    char as[10]="cdefgabC";
    char ds[10]="Cbagfedc";

    printf("음 입력\n");
    scanf("%s",ss);
    printf("\n");

    printf("결과 : ");
    int sw=pp(ss,as,ds);

    switch (sw) {
        case 1:
            printf("ascending");
            break;
        case 2:
            printf("descending");
            break;
        case 3:
            printf("mixed");
            break;
        default:
            break;
    }
    printf("\n");

    return 0;
}
```

->필요한 변수선언과 입력받아주고 switch-case문을 활용했습니다. return값을 이용했습니다.

<비교하는 함수>

```
int pp(char ss[10],char as[10],char ds[10]){
    if(compare(ss,as)==0){
        return 1;
    }
    else if (compare(ss,ds)==0){
        return 2;
    }
    else{
        return 3;
    }
    return 0;
}
```

->비교해서 return값을 메인함수의 switch-case문에 주는 것을 함수화해서 사용했습니다.

<strcmp함수>

```
//strcmp함수
int compare(const char str1[], const char str2[]) {
    int i = 0;

    while (str1[i] && str1[i] == str2[i]) {
        i++;
    }

    return (unsigned char)str1[i] - (unsigned char)str2[i];
}
```

->string.h안에 있는 strcmp 구현해서 사용했습니다.

출력 결과

```
음 입력
cdefgabcC

결과 : ascending
Program ended with exit code: 0
```

4.

찾음을 당할 문자열과 찾을 문자열을 선언하고 찾을 문자열을 찾을 당할 문자열에 대해 비교하며 구현했습니다. 복잡한 내용이라 설명이 좀 빈약하다는 생각이 듭니다. 양해 부탁드립니다.

<메인 함수>

```
int main(int argc, const char * argv[]) {
    //변수 선언
    char inputStr[100], subStr[100];
    int index = 0;

    //입력 받기
    printf("최대 99개 문자 입력 <inputStr> : ");
    scanf("%s", inputStr);
    printf("찾는 문자열 <subStr> : ");
    scanf("%s", subStr);

    while(1){
        int temp = compare(inputStr, subStr, index);
        if(temp == 0){
            break;
        }
        printf("%s의 위치 : %d\n", subStr, temp);
        index = temp + len(subStr);
    }

    return 0;
}
```

->필요한 변수선언 및 입력 처리해주었습니다. 무한반복안에 같은것을 찾는 compare함수를 선언했습니다. 0일때 break건 이유는 함수 내에서 몇번째에 같은게 있는지를 return해서 temp에 저장하게 할때 0이 return 됐다는 것은 같은게 아예 없거나 더이상 없다는 의미이기 때문입니다. 같은 문자열을 찾은 후 그 이후부터 비교하기 위해 index변수를 사용해 저장했습니다.

<같은 문자열 찾는 함수>

```
int compare(char inputStr[], char subStr[], int index){
    for(int i = index; i < len(inputStr) - len(subStr) + 1; i++){
        for(int j = 0; j < len(subStr); j++){
            if(subStr[j] == inputStr[i+j] && j == len(subStr)-1){
                return i+1;
            }
            else if(subStr[j] != inputStr[i+j]){
                break;
            }
        }
    }
    return 0;
}
```

->맨 앞에 수를 비교하고 같으면 다음까지 비교한 후 다 같다면 리턴하게 했습니다. 리턴한 후 뒤에도 같은게 있을 수 있으니 리턴시킨 그 자리에서부터 다시



입력 받아서 비교하게 하는 함수를 구현해서 사용했습니다.

<strlen 함수>

```
int len(char str[]){
    for (int i=0; i<100; i++) {
        if(str[i]=='\0'){
            return i;
        }
    }
    return 0;
}
```

->string.h에 있는 strlen 함수 구현해서 사용했습니다.

출력 결과

```
최대 99개 문자 입력 <inputStr> : asdfgdfghjdf
찾는 문자열 <subStr> : df
df의 위치 : 3
df의 위치 : 6
df의 위치 : 11
Program ended with exit code: 0
```

5.

문제 이해를 제대로 한지 확신이 안들지만.. 일단 풀었습니다. 자연수 N개를 입력 받는데 이게 홀수여야하고 중앙값은 크기순으로 정렬했을때 가운데 있는 숫자니깐 홀수개여야 하는게 중앙값 생겨야하니깐으로 이해했습니다. 배열을 입력 받고 크기순으로 나열한 후 입력받은 숫자를 중앙값으로 하는 수열의 부분수열의 개수를 구하는 알고리즘을 구현했습니다.

<메인 함수>

```

int main(int argc, const char * argv[]) {
    //변수 선언
    int n,b,s,s1,tmp;
    int nn[100]={0,};

    //n값과 b값 입력
    printf("N : ");
    scanf("%d",&n);
    printf("B : ");
    scanf("%d",&b);
    printf("\n");

    //약수일때 예외처리
    if(n%2==0){
        printf("약수입니다. 홀수를 입력해주세요.");
        return 0;
    }

    //수열 입력
    for (int i=0; i<n; i++) {
        scanf("%d",&nn[i]);
    }

    //중앙
    for (int i=0; i<n-1; i++) {
        for (int j=i+1; j<n; j++) {
            if(nn[i]>nn[j]){
                tmp=nn[i];
                nn[i]=nn[j];
                nn[j]=tmp;
            }
        }
    }

    //중앙값이 몇번째 수열에 있는지 찾기
    for (int i=0; i<n; i++) {
        if(b==nn[i]){
            s=i; //5
            break;
        }
    }

    //예외 부분수열의 개수 7-4=3
    if(s1>=(n+1)/2){
        s=n-s1;
        printf("중앙값이 %d인 부분수열의 개수 : %d\n",b,s);
    }
    else if(s1<=(n+1)/2){
        s=s1;
        printf("중앙값이 %d인 부분수열의 개수 : %d\n",b,s);
    }

    return 0;
}

```

-> 일단 n값을 받아서 그만큼 배열을 입력받았습니다. 그 후 순서대로 정렬했습니다. 부분수열을 구해낼때 입력받은 n값을 가지고 중앙이 몇번째 배열인지 알아낸 후 그를 기준으로 클때와 작을때로 나눴습니다. 작으면 b값이 부분수열의 개수이고 크거나 같으면 n에서 중앙에 있는 값을 빼서 부분 수열을 구현했습니다.

출력 결과

```

N : 7
B : 4

5 7 2 4 3 1 6
중앙값이 4인 부분수열의 개수 : 3
Program ended with exit code: 0

```