

## CHAPTER 12

# 스택 구현 및 응용

# 실습 1

문제 1)

입력 받은 문자열을 뒤집어서 출력해보자.  
(문자열과 스택의 크기는 100 이하)

```
입력 : Hello World!  
출력 : !dlroW olleH
```

# 실습 1 정답

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

#define max 100

char stack[max];
int top = 0;
void main() {
    printf("입력 : ");
    char str[100]={NULL,};

    scanf("%[^\n]s", str);
    for (int i = 99; i >= 0; i--) {
        if (str[i] == NULL)continue;
        stack[top++] = str[i];
    }
    printf("출력 : %s\n", &stack);
}
```

## 실습 2

문제 2)

정수 하나를 입력 받고, 이진법으로 표현한 것을 거꾸로 출력하는 프로그램을 만들어보자.

```
입력 : 20  
출력 : 00101
```

# 실습 2 정답

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

#define max 100

char stack[max];
int top = 0;
void main() {
    printf("입력 : ");
    int num;
    scanf("%d", &num);

    while (num) {
        stack[top++] = num % 2 + '0';
        num /= 2;
    }

    printf("출력 : %s\n", &stack);
}
```

## 실습 3

### 문제 3)

‘+’를 입력한 후 정수를 입력 받으면 스택에 넣고,

‘-’를 입력하면 스택의 맨 위 요소를 출력하고 스택에서 제거한다.

‘\*’을 입력하면 프로그램을 종료한다.

스택이 가득 찬 상태에서 ‘+’를 입력하면 스택에 있는 모든 수를 출력한다.

이때, 정수의 크기는 0 이상 9 이하, 스택의 크기는 5.

스택이 빈 상태에서는 아무런 동작을 하지 않는다.

```
명령어 입력(+,-,*) : 1
ERROR!
명령어 입력(+,-,*) : +
정수 입력 : 1
명령어 입력(+,-,*) : +
정수 입력 : 2
명령어 입력(+,-,*) : -
정수 제거 및 출력 : 2
명령어 입력(+,-,*) : +
정수 입력 : 3
명령어 입력(+,-,*) : +
정수 입력 : 4
명령어 입력(+,-,*) : +
정수 입력 : 5
명령어 입력(+,-,*) : +
정수 입력 : 6
명령어 입력(+,-,*) : +
13456
종료되었습니다.
```

```
명령어 입력(+,-,*) : +
정수 입력 : 1
명령어 입력(+,-,*) : *
종료되었습니다.
```

# 실습 3 정답

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

```
#define max 5
```

```
char stack[max];
```

```
int top = 0;
```

```
void main() {
```

```
    char cmd;
```

```
    while (1) {
```

```
        printf("명령어 입력(+,-,*) :");
```

```
        scanf("%c", &cmd);
```

```
        getc(stdin);
```

```
        if (cmd == '+') {
```

```
            if (top == max) {
```

```
                printf("%s\n", stack);
```

```
                printf("종료되었습니다.\n");
```

```
                return;
```

```
            }
```

```
            int n;
```

```
            printf("정수 입력 : ");
```

```
            scanf("%d", &n);
```

```
            getc(stdin);
```

```
            stack[top++] = n + '0';
```

```
        }
```

```
        stack[top++] = n + '0';
```

```
    }
```

```
    else if (cmd == '-') {
```

```
        printf("정수 제거 및 출력 : %c\n", stack[top-1]);
```

```
        stack[top--] = NULL;
```

```
    }
```

```
    else if (cmd == '*') {
```

```
        printf("종료되었습니다.\n");
```

```
        return;
```

```
    }
```

```
    else {
```

```
        printf("ERROR!\n");
```

```
    }
```

```
}
```

```
printf("출력 : %s\n", &stack);
```

```
}
```

## 실습 4

문제 4)

입력할 정수의 개수를 입력하고, 해당 수의 합을 구한다.  
이때, 입력한 정수가 0일 때, 최근에 입력한 수를 제거한다.

```
입력할 수의 개수 : 4
정수 : 33
정수 : 0
정수 : 4
정수 : 0
합 : 0
```

```
입력할 수의 개수 : 10
정수 : 1
정수 : 3
정수 : 5
정수 : 4
정수 : 0
정수 : 0
정수 : 7
정수 : 0
정수 : 0
정수 : 6
합 : 7
```



## 실습 4 정답

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#define MAX 100000
```

```
int stack[MAX];
int top = -1;
```

```
void push(int data) {
    if (top >= MAX - 1) return;
    stack[++top] = data;
}
```

```
void pop() {
    if (top < 0) return;
    stack[top--] = 0;
}
```

```
int main() {
    int K, data, sum = 0;
    printf("입력할 수의 개수 : ");
    scanf("%d", &K);
```

```
    //데이터 입력
    while (K-- > 0) {
        printf("정수 : ");
        scanf("%d", &data);
        if (data == 0) pop();
        else push(data);
    }
```

```
    //최종 합 구하기
    for (int i = 0; i <= top; i++)
        sum += stack[i];
    printf("합 : %d\n", sum);
    return 0;
}
```

## 실습 5

문제 5)

괄호가 짝을 이루는지 확인해보자

```
6
(())()
NO
(((())())
NO
(())(())
YES
((())(())(())())
NO
()()()()()()()
YES
(()(())()
NO
```

# 실습 5 정답

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

int main()
{
    int inputNumber;
    scanf("%d", &inputNumber);
    char bracket[50];
    for (int i = 0; i < inputNumber; i++) {
        int cnt = 0;
        scanf("%s", bracket);
        for (int j = 0; j < strlen(bracket); j++) {
            if (bracket[j] == '(') {
                cnt++;
            }
            else {
                cnt--;
            }
            if (cnt < 0) {
                printf("NO\n");
                break;
            }
        }
        if (cnt == 0) {
            printf("YES\n");
        }
        else if (cnt > 0) {
            printf("NO\n");
        }
    }
    return 0;
}
```

## 실습 6

문제 6)

짝수만 담은 스택을 만들어보고, 출력해보자.

이때, 입력하는 정수는 9 이하의 정수.

```
입력할 정수의 개수 : 10
수 : 1
수 : 2
수 : 3
수 : 4
수 : 5
수 : 6
수 : 7
수 : 8
수 : 9
수 : 7
{8, 6, 4, 2}
```

# 실습 6 정답

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

char stack[51];
int top = 0;

int main()
{
    int n;
    printf("입력할 정수의 개수 : ");
    scanf("%d", &n);

    int num;

    while (n-- > 0) {
        printf("정수 : ");
        scanf("%d", &num);
        if (num < 10 && num % 2 == 0) stack[top++] = num + '0';
    }
    printf("{");

    while (top > 0) {
        printf("%c", stack[top]);
        if (top > 1) printf(",");
    }
    printf("}\n");
}
```

## 실습 7

문제 7)

정수를 입력했을 때, 거꾸로 뒤집은 숫자와 비교하여 크면 'bigger'  
작으면 'smaller', 같으면 'equal'을 출력하는 프로그램을  
만들어보자

```
1234  
smaller
```

```
2002  
equal
```

```
4321  
bigger
```

# 실습 7 정답

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

int stack[51];
int top = 0;

int main()
{
    int n;
    scanf("%d", &n);
    int t = n;
    int num=0;

    while (n) {
        stack[top++] = n % 10;
        n /= 10;
    }
    int tt = 1;
    for (int i = top-1; i >=0; i--) {
        num += tt * stack[i];
        tt *= 10;
    }
    if (t > num)printf("bigger");
    else if (t < num)printf("smaller");
    else printf("equal");
}
```

## 실습 8

문제 8)

공백 포함 문자열을 입력 받아서 한 단어씩 뒤집어보자!

```
computer information engineering  
retupmoc gnireenigne noitamrofni
```



# 실습 8 정답

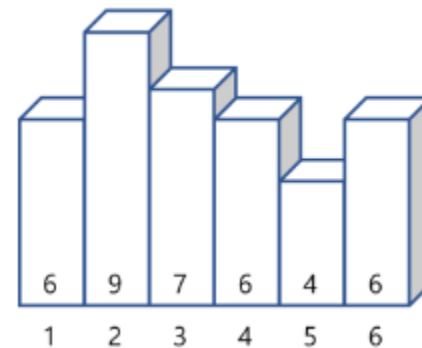
```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
char stack[100001];
int top = 0;

int main()
{
    char n[100001];
    scanf("%[^\\n]s", n);

    char* str;
    str = strtok(n, " ");
    while (str) {
        int i = strlen(str);
        while (i-- > 0) {
            stack[top++] = str[i];
        }
        printf("%s ", stack);
        memset(stack, NULL, sizeof(stack) * sizeof(char));
        top = 0;
        str = strtok(NULL, "\\n");
    }
}
```

## 실습 9

문제 9)  
n개의 막대기가 있을 때,  
오른쪽에서 보아서 몇개가 보이는지  
알아내보자.



```
총 막대기의 개수 : 6
막대기의 길이 : 6
막대기의 길이 : 9
막대기의 길이 : 7
막대기의 길이 : 6
막대기의 길이 : 4
막대기의 길이 : 6
보이는 막대기의 개수는 : 3
```

# 실습 9 정답

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

int main(void)
{
    int arr[100005];

    int n;
    int cnt = 1;
    int top = 0;
    printf("총 막대기의 개수 : ");
    scanf("%d", &n);

    for (int i = 0; i < n; ++i) {
        printf("막대기의 길이 : ");
        scanf("%d", &arr[i]);
    }

    top = arr[n - 1];
    for (int i = n - 1; i >= 0; --i)
    {
        if (arr[i] > top)
        {
            cnt++;
            top = arr[i];
        }
    }

    printf("보이는 막대기의 개수는 : %d", cnt);
}
```

## 실습 10

문제 10)

괄호 안의 문자를 제외한 나머지를 뒤집어보자.  
이때 문자열 내에 공백은 없다고 가정한다.

```
(aespa)newjeans(idle)
(aespa)snaejwen(idle)
```

# 실습 10 정답

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

int main(void)
{
    char arr[100005];

    char input[100001];

    int top = 0;
    scanf("%s", input);

    int status = 0;
    int i = 0;
    while (input[i]) {
        if (input[i] == '(') {
            for (int j = top - 1; j >= 0; j--) {
                printf("%c", arr[j]);
            }
            top = 0;
            status = 1;
        }
        if (status) {
            printf("%c", input[i]);
        }
        else {
            arr[top++] = input[i];
        }
        if (input[i] == ')') {
            status = 0;
        }
        i++;
    }
}
```

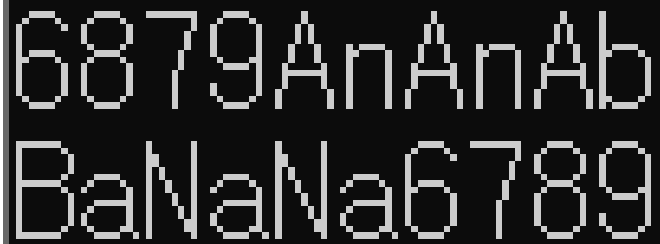
## 실습 11

문제 11)

단어를 모두 뒤집어보자!

이때, 영어 소문자는 대문자로, 대문자는 소문자로 바꾼다.  
그리고 숫자 6은 9로, 숫자 9는 6으로 바꾼다.

예시 : BaNaNa6789 -> 6879AnAnAb



6879AnAnAb  
BaNaNa6789

# 실습 11 정답

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

int main(void)
{
    char arr[100005];

    char input[100001];

    int top = 0;
    scanf("%s", input);

    int i = 0;
    while (input[i]) {
        arr[top++] = input[i++];
    }
    while (top--) {
        if (arr[top] == '6') printf("9");
        else if (arr[top] == '9') printf("6");
        else if (arr[top] >= 'a' && arr[top] <= 'z') {
            char temp = arr[top] - 32;
            printf("%c", temp);
        }
        else if (arr[top] >= 'A' && arr[top] <= 'Z') {
            char temp = arr[top] + 32;
            printf("%c", temp);
        }
        else printf("%c", arr[top]);
    }
}
```

## 실습 12

문제 12)

괄호와 괄호 안의 단어를 한 글자로 취급할 때,  
문자열을 뒤집어 출력하는 프로그램을 만들어보자

예시 : ab(cde)fg -> gf(cde)ba

```
ive(blackpink)itzy  
yzt i(blackpink)evi
```



# 실습 12 정답

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

char temp[100][100001];
int main(void)
{
    char arr[100005];

    char input[100001];

    int top = 0;
    scanf("%s", input);

    int idx = 0;
    int top2 = 0;
    for (int i = 0; i < strlen(input); i++) {
        if (input[i] == '(') {
            char tt[100001];
            int a = 0;
            for (; input[i] != ')'; i++) {
                tt[a] = input[i];
                a++;
            }
            tt[a] = '\0';
            for (int k = 0; k < strlen(tt); k++) {
                temp[top2][k] = tt[k];
            }
            arr[top++] = '*';

            top2++;
        }
        else {
            arr[top++] = input[i];
        }
    }
    while (top--) {
        if (arr[top] == '*') {
            top2--;
            for (int i = 0; temp[top2][i] != '\0'; i++) {
                printf("%c", temp[top2][i]);
            }
        }
        else {
            printf("%c", arr[top]);
        }
    }
}
```

## 실습 13

문제 13)

정수 10개를 스택에 입력 받고, 최대값을 구하는 프로그램을 만들어보자.

```
8
2
4
1
5
3
2
91
4
6
최대 : 91
```

# 실습 13 정답

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

int main(void)
{
    int arr[100005];

    int top = 0;

    while (top != 10) {
        int n;
        scanf("%d", &n);
        arr[top++] = n;
    }

    int max = 0;
    while (top-- > 0) {
        max = max < arr[top] ? arr[top] : max;
    }
    printf("최대 : %d", max);
}
```

## 실습 14

문제 14)

문자열과 폭발 문자열을 입력받는다.

처음 문자열 내에 폭발 문자열이 폭발하고 남은 문자열은 합쳐진다.

폭발 문자열이 없을 때까지 반복된다.

남은 문자열이 있는 경우 문자열을 출력하고, 문자열이 없는 경우 'END'를 출력한다.

```
hhelloello  
hello  
END
```

```
abaababbababbbb  
ab  
bb
```

# 실습 14 정답

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>
#include <stdbool.h>

char origin[1000001], stack[1000001], ex[37];
void main()
{
    int j, sIndex, exLen, tempJ, tempJ;
    bool isSame;
    scanf("%s%s", origin, ex);
    exLen = strlen(ex) - 1;
    for (int i = sIndex = 0; origin[i]; i++)
    {
        stack[sIndex++] = origin[i];
        if (stack[sIndex - 1] == ex[exLen])
        {
            isSame = true;
            for (j = exLen, tempJ = 1; j >= 0; j--)
                if (sIndex < exLen + 1 || ex[j] != stack[sIndex - tempJ++])
                {
                    isSame = false;
                    break;
                }
            if (isSame) stack[sIndex -= exLen + 1] = 0;
        }
    }
    if (stack[sIndex - 1] == ex[exLen])
    {
        isSame = true;
        for (j = exLen, tempJ = 1; j >= 0; j--)
            if (sIndex - 1 < exLen || ex[j] != stack[sIndex - tempJ++])
            {
                isSame = false;
                break;
            }
        if (isSame) stack[sIndex -= exLen + 1] = 0;
    }
    stack[sIndex] = 0;
    puts(stack[0] == 0 ? "END" : stack);
}
```

## 실습 15

문제 15)

회문(거꾸로 읽어도 똑같은 문장)을 찾는 프로그램을 스택을 사용해서 구현해보자!

회문 예시 : level, mom, 등

```
newJeans  
회문이 아닙니다.
```

```
level  
회문이 맞습니다.
```

# 실습 15 정답

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

int main(void)
{
    char arr[100005];

    char input[100001];

    int top = 0;
    scanf("%s", input);

    int i = 0;
    while (input[i]) {
        arr[top++] = input[i++];
    }
    i = 0;
    int status = 0;
    while (top--) {
        if (arr[top] != input[i++]) {
            status = 1;
            break;
        }
    }
    if (!status)printf("회문이 맞습니다.\n");
    else printf("회문이 아닙니다.\n");
}
```

## 실습 16

문제 16)

문자열을 입력했을 때, 해당 문자열 내에 같은 문자가 연속되지 않도록 저장하고, 거꾸로 출력해보자!

```
yyoobbbbbepppyyh  
hypeboy
```



# 실습 16 정답

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

int main(void)
{
    char arr[100005];

    char input[100001];

    int top = 0;
    scanf("%s", input);

    int i = 0;
    char temp='\\n';
    while (input[i]) {
        if (temp != input[i]) {
            temp = input[i];
            arr[top++] = input[i];
        }
        i++;
    }
    while (top-->0) {
        printf("%c",arr[top]);
    }
}
```

## 실습 17

문제 17)

스택을 이용해서 가장 작은 숫자를 구해보자

```
5
23 243 43 12 5
5
```

# 실습 17 정답

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<string.h>
#include<stdlib.h>

int main(void)
{
    int arr[100005];

    int n;
    int temp = 0;
    scanf("%d", &n);

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

    temp = 1000000;
    for (int i = n - 1; i >= 0; --i)
    {
        if (arr[i] < temp)
        {
            temp = arr[i];
        }
    }

    printf("%d", temp);
}
```

## 실습 18

문제 18)

후위표기식으로 표현된 수식을 계산해보자.

여기서 피연산자는 알파벳 대문자만 사용되며, 수식을 입력한 이후, 피연산자의 개수만큼 숫자를 입력받는다.

```
5
ABC*+DE/-
1
2
3
4
5
6.20
```

```
1
AA+A+
1
3.00
C++
```

# 실습 18 정답

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <ctype.h>
#define TRUE      1
#define FALSE     0
```

```
typedef double Data;
```

```
typedef struct _node
{
    Data data;
    struct _node* next;
} Node;
```

```
typedef struct _listStack
{
    Node* head;
} ListStack;
```

```
typedef ListStack Stack;
void StackInit(Stack* pstack)
{
    pstack->head = NULL;
}
```

```
int SIsEmpty(Stack* pstack)
{
    if (pstack->head == NULL)
        return TRUE;
    else
        return FALSE;
}

void SPush(Stack* pstack, Data data)
{
    Node* newNode = (Node*)malloc(sizeof(Node));

    newNode->data = data;
    newNode->next = pstack->head;

    pstack->head = newNode;
}
```

```
Data SPop(Stack* pstack)
{
    Data rdata;
    Node* rnode;

    if (SIsEmpty(pstack))
    {
        printf("Stack Memory Error!");
        exit(0);
    }

    rdata = pstack->head->data;
    rnode = pstack->head;

    pstack->head = pstack->head->next;
    free(rnode);

    return rdata;
}
```

```
Data SPeek(Stack* pstack)
{
    if (SIsEmpty(pstack))
    {
        printf("Stack Memory Error!");
        exit(0);
    }

    return pstack->head->data;
}
```

# 실습 18 정답

```
double EvalRPNExp(char exp[], int arr[])
{
    Stack stack;
    int expLen = strlen(exp);
    int i;
    int tok;
    double op1, op2;

    StackInit(&stack);

    for (i = 0; i < expLen; i++)
    {
        tok = exp[i];

        if (tok > 64 && tok < 91)
        {
            SPush(&stack, (double)arr[tok - 65]);
        }
        else
        {
            op2 = SPop(&stack); // 먼저 꺼낸 값이 두 번째 피연산자!
            op1 = SPop(&stack);

            switch (tok)
            {
            case '+':
                SPush(&stack, op1 + op2);
                break;
            case '-':
                SPush(&stack, op1 - op2);
                break;
            case '*':
                SPush(&stack, op1 * op2);
                break;
            case '/':
                SPush(&stack, op1 / op2);
                break;
            }
        }
    }
    return SPop(&stack);
}
```

```
int main(void)
{
    int operatredNum, i, len;
    char postfix[200];
    int alphaNum[30];
    scanf("%d", &operatredNum);
    scanf("%s", postfix);
    len = strlen(postfix);
    for (i = 0; i < operatredNum; i++) {
        scanf("%d", &alphaNum[i]);
    }
    printf("%.2lf", EvalRPNExp(postfix, alphaNum));
    return 0;
}
```

## 실습 19

문제 19)

괄호가 올바르게 되도록 하기 위해서 붙여야 할 괄호의 최소 개수를 구해보자.

)))(((  
6

(((((  
4

# 실습 19 정답

```
#define _CRT_SECURE_NO_WARNINGS
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<stdbool.h>
#include<math.h>
#define MAX_SIZE 53
typedef struct Intstack {
    int size;
    char stack[MAX_SIZE];
}Intstack;

Intstack s;
char str[MAX_SIZE];

void Init_Stack(Intstack* s) {
    s->size = 0;
}

int Push(Intstack* s, char x) {
    if (s->size + 1 > MAX_SIZE) return 0;

    s->stack[s->size++] = x;
    return 1;
}

char Pop(Intstack* s) {
    if (s->size <= 0) return 0;
    else return s->stack[--s->size];
}
```

```
char Top(Intstack* s) {
    if (s->size <= 0) return 0;
    else return s->stack[s->size - 1];
}

int Size(Intstack* s) {
    return s->size;
}

void Solve(int len) {
    for (int i = 0; i < len; i++) {
        if (str[i] == '(') {
            Push(&s, str[i]);
        }
        else {
            if (Top(&s) == '(') {
                Pop(&s);
            }
            else {
                Push(&s, str[i]);
            }
        }
    }
}
```

```
int main() {
    Init_Stack(&s);
    scanf("%s", str);
    int length = strlen(str);
    Solve(length);

    printf("%d\n", Size(&s));

    return 0;
}
```



## 실습 20

문제 20)

동작을 할 횟수를 입력하고, 입력한 횟수만큼 스택에 문자를 넣는다. 이때, 0을 입력하면 가장 최근에 입력한 문자가 제거된다. 스택이 비어있으면 YES를, 비어있지 않다면 NO를 출력한다.

```
4
a
0
b
0
YES
```

```
4
0
0
0
0
YES
```

# 실습 20 정답

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <string.h>
#include <stack>
using namespace std;

int n, ans, len;
char buf;

int main() {
    scanf("%d%c", &n);
    int top = 0;
    char arr[100001];
    for (int i = 0; i < n; ++i) {
        scanf("%c%c", &buf);
        if (buf=='0') {
            if (top <= 0)continue;
            arr[top--] = '\0';
            continue;
        }
        arr[top++] = buf;
    }
    if(top)
        printf("NO");
    else
        printf("YES");
}
```