Programming Examples

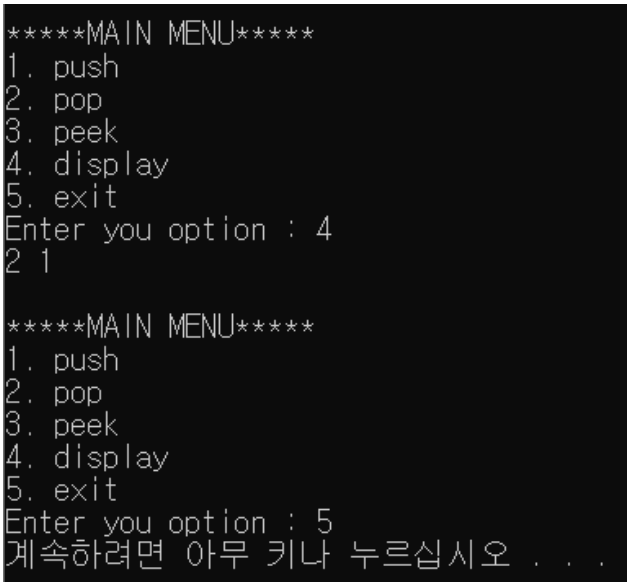
1.

```
#include <stdio.h>
# define max 3
void display(int a[]);
void push(int a[],int x);
int peek(int a[]);
int pop(int a[]);
int top=-1;
int a[max];
int main()
{
        int option,x;
        do
        {
                printf("\n\n");
                printf("*****MAIN MENU*****\n");
                printf("1. push\n");
                printf("2. pop\n");
                printf("3. peek\n");
                printf("4. display\n");
                printf("5. exit\n");
                printf("Enter you option : ");
                scanf("%d",&option);
                switch(option)
                {
                case 1:
                        printf("Enter the number to be pushed on stack: ");
                        scanf("%d",&x);
                        push(a,x);
                        break;

                case 2:
                        x=pop(a);
                        if(x!=-1)
                                printf("The value deleted form stack is : %d",x);
                        break;

                case 3:
                        x=peek(a);
                        if(x!=-1)
                                printf("The value stored at top of stack is : %d",x);
                                break;
```

```
                case 4:
                        display(a);
                        break;
                }
        }while(option!=5);
}

void push(int a[],int x)
{
        if(top==max-1)
                printf("stack overflow");
        else
        {
                top++;
                a[top]=x;
        }
}

int pop(int a[])
{
        int x;
        if(top==-1)
                {
                        printf("stack underflow");
                        return -1;
                }
        else
        {
                x=a[top];
                top--;
                return x;
        }
}

int peek(int a[])
{
        if(top==-1)
        {
                printf("stack is empty");
                return -1;
        }
        else
                return(a[top]);
```

```
}

void display(int a[])
{
        int i;
        if(top==-1)
                printf("stack is empty");
        else
        {
                for(i=top;i>=0;i--)
                        printf("%d ",a[i]);

        }
}
```

```
*****MAIN MENU*****
1. push
2. pop
3. peek
4. display
5. exit
Enter you option : 1
Enter the number to be pushed on stack: 1


*****MAIN MENU*****
1. push
2. pop
3. peek
4. display
5. exit
Enter you option : 1
Enter the number to be pushed on stack: 2


*****MAIN MENU*****
1. push
2. pop
3. peek
4. display
5. exit
Enter you option : 1
Enter the number to be pushed on stack: 3


*****MAIN MENU*****
1. push
2. pop
3. peek
4. display
5. exit
Enter you option : 2
The value deleted form stack is : 3

*****MAIN MENU*****
1. push
2. pop
3. peek
4. display
5. exit
Enter you option : 3
The value stored at top of stack is : 2
```



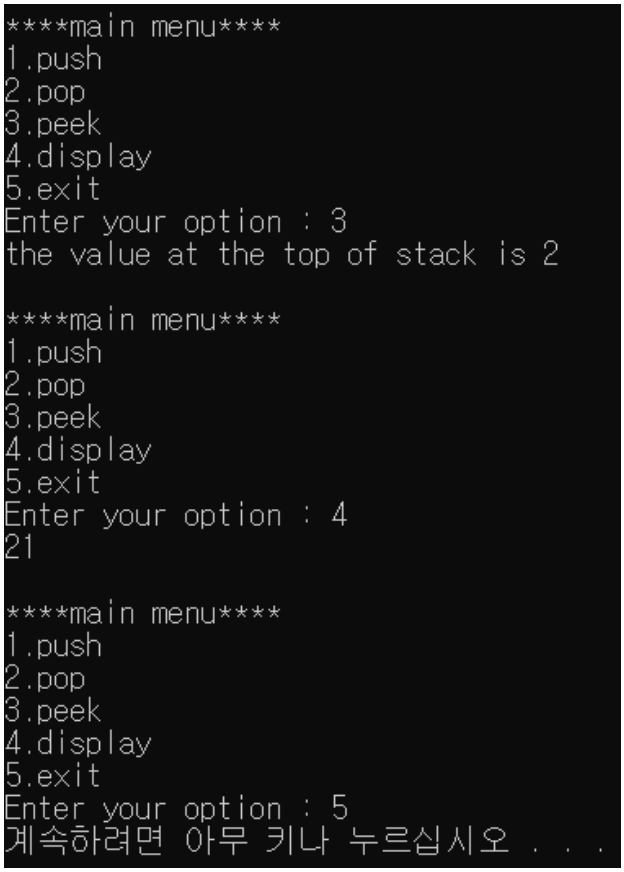
2.

```
#include <stdio.h>
#include <malloc.h>
struct stack
{
        int data;
        struct stack *next;
};
struct stack *push(struct stack *top, int x);
struct stack *pop(struct stack *top);
int peek(struct stack *top);
struct stack *display(struct stack *top);
struct stack *top=NULL;

int main()
{
        int x,option;
        do{
                printf("\n\n");
                printf("****main menu****\n");
                printf("1.push\n");
                printf("2.pop\n");
                printf("3.peek\n");
                printf("4.display\n");
                printf("5.exit\n");
                printf("Enter your option : ");
                scanf("%d",&option);
                switch(option)
                {
```

```
                case 1:
                        printf("Enter the number to be pushed on the stack : ");
                        scanf("%d",&x);
                        top=push(top,x);
                        break;
                case 2:
                        top=pop(top);
                        break;
                case 3:
                        x=peek(top);
                        if(x==-1)
                                printf("stack is empty");
                        else
                                printf("the value at the top of stack is %d",x);
                        break;
                case 4:
                        top=display(top);
                        break;
                }

        }while(option!=5);
}

struct stack *push(struct stack *top, int x)
{
        struct stack *new_node;
        new_node=(struct stack*)malloc(sizeof(struct stack));
        new_node->data=x;
        if(top==NULL)
        {
                new_node->next=NULL;
                top=new_node;
        }
        else
        {
                new_node->next=top;
                top=new_node;
        }
        return top;
}

struct stack *pop(struct stack *top)
{
```

```
        struct stack *ptr;
        ptr=top;
        if(top==NULL)
                printf("stack underflow");
        else
        {
                top=top->next;
                printf("The value being deleted is : %d",ptr->data);
                free(ptr);
        }
        return top;
}

int peek(struct stack *top)
{
        if(top==NULL)
                return -1;
        else
                return top->data;
}

struct stack *display(struct stack *top)
{
        struct stack *ptr;
        ptr=top;
        if(top==NULL)
                printf("stack is empty");
        else
        {
                while(ptr!=NULL)
                {
                        printf("%d",ptr->data);
                        ptr=ptr->next;
                }
        }
        return top;
}
```

```
****main menu****
1.push
2.pop
3.peek
4.display
5.exit
Enter your option : 1
Enter the number to be pushed on the stack : 1


****main menu****
1.push
2.pop
3.peek
4.display
5.exit
Enter your option : 1
Enter the number to be pushed on the stack : 2


****main menu****
1.push
2.pop
3.peek
4.display
5.exit
Enter your option : 1
Enter the number to be pushed on the stack : 3


****main menu****
1.push
2.pop
3.peek
4.display
5.exit
Enter your option : 4
321

****main menu****
1.push
2.pop
3.peek
4.display
5.exit
Enter your option : 2
The value being deleted is : 3
```



3.

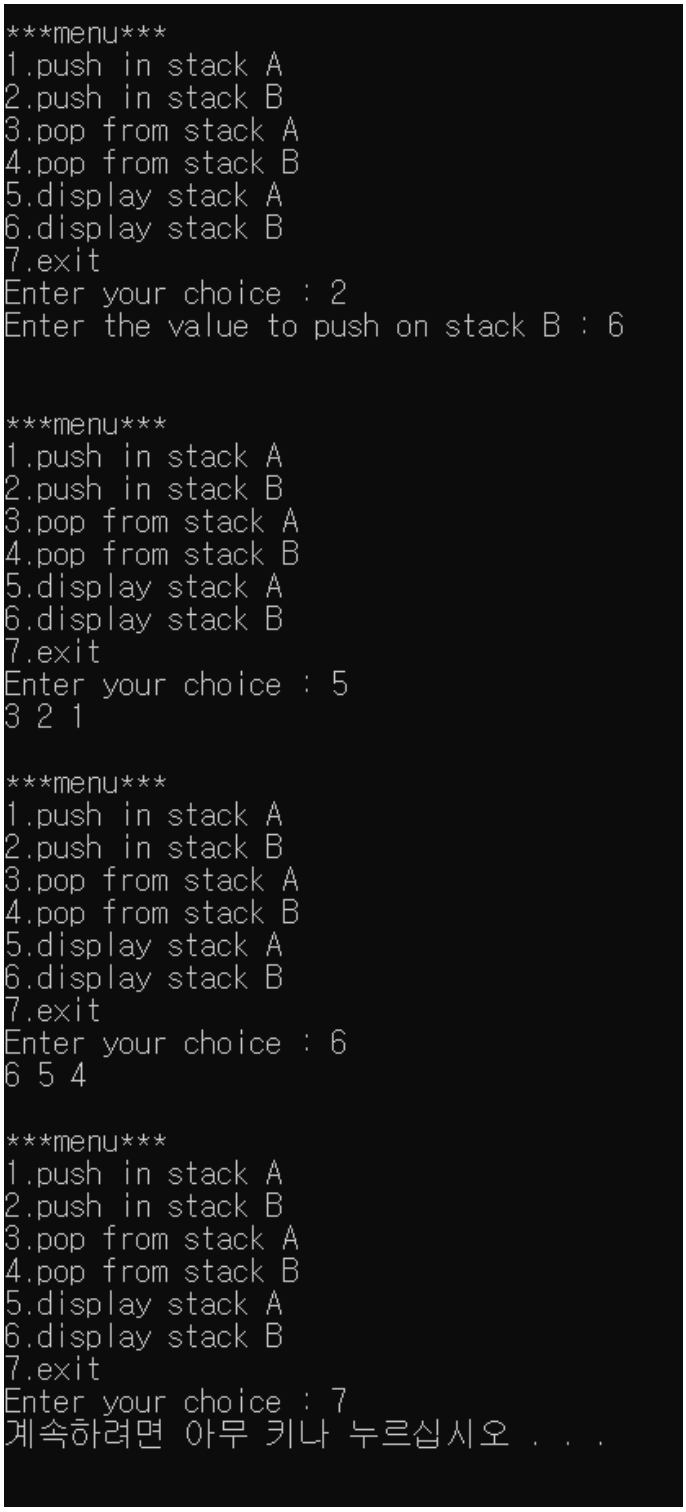
```
#include <stdio.h>
#define max 10
int S[max], topA=-1, topB=max;
void push_A(int x);
void push_B(int x);
int pop_A();
int pop_B();
void display_A();
void display_B();
int main()
{
        int option;
        int x;
        do
        {
                printf("\n\n");
                printf("***menu***\n");
                printf("1.push in stack A\n");
                printf("2.push in stack B\n");
                printf("3.pop from stack A\n");
```

```
                printf("4.pop from stack B\n");
                printf("5.display stack A\n");
                printf("6.display stack B\n");
                printf("7.exit\n");
                printf("Enter your choice : ");
                scanf("%d",&option);
                switch(option)
                {
                case 1:
                        printf("Enter the value to push on stack A : ");
                        scanf("%d",&x);
                        push_A(x);
                        break;

                case 2:
                        printf("Enter the value to push on stack B : ");
                        scanf("%d",&x);
                        push_B(x);
                        break;

                case 3:
                        x=pop_A();
                        if(x!=-1)
                                printf("The value popped from stack A= %d",x);
                        break;

                case 4:
                        x=pop_B();
                        if(x!=-1)
                                printf("The value popped from stack B=%d",x);
                        break;

                case 5:display_A();
                        break;
                case 6:display_B();
                        break;

                }


        }while(option!=7);
}
```

```
void push_A(int x)
{
        if(topA==topB-1)
                printf("stack overflow");
        else
                {
                        topA=topA+1;
                        S[topA]=x;
                }

}

void push_B(int x)
{
        if(topA==topB-1)
                printf("stack overflow");
        else
        {
                topB=topB-1;
                S[topB]=x;
        }
}

int pop_A()
{
        int x;
         if(topA==-1)
         {
                  printf("stack underflow");
         }
         else
         {
                x=S[topA];
                topA--;
         }
         return x;
}

int pop_B()
{
        int x;
        if(topB==max)
        {
```

```
                printf("stack underflow");
        }
        else
        {
                x=S[topB];
                topB++;
        }
        return x;
}

void display_A()
{
        int i;
        if(topA==-1)
                printf("stack is empty");
        else
        {
                for(i=topA;i>=0;i--)
                        printf("%d ",S[i]);
        }
}

void display_B()
{
        int i;
        if(topB==max)
                printf("stack is empty");
        else
        {
                for(i=topB;i<max;i++)
                        printf("%d ",S[i]);
        }
}
```

```
***menu***
1.push in stack A
2.push in stack B
3.pop from stack A
4.pop from stack B
5.display stack A
6.display stack B
7.exit
Enter your choice : 1
Enter the value to push on stack A : 1


***menu***
1.push in stack A
2.push in stack B
3.pop from stack A
4.pop from stack B
5.display stack A
6.display stack B
7.exit
Enter your choice : 1
Enter the value to push on stack A : 2


***menu***
1.push in stack A
2.push in stack B
3.pop from stack A
4.pop from stack B
5.display stack A
6.display stack B
7.exit
Enter your choice : 1
Enter the value to push on stack A : 3


***menu***
1.push in stack A
2.push in stack B
3.pop from stack A
4.pop from stack B
5.display stack A
6.display stack B
7.exit
Enter your choice : 2
Enter the value to push on stack B : 4


***menu***
1.push in stack A
2.push in stack B
3.pop from stack A
4.pop from stack B
5.display stack A
6.display stack B
7.exit
Enter your choice : 2
Enter the value to push on stack B : 5
```



4.

```
#include <stdio.h>
#define max 5
int top=-1;
int S[max];
void push(int a);
```

```
int pop();
int main()
{
	int arr[8];
	int num,i,x;
	printf("Enter the number of elements in the array : ");
	scanf("%d",&num);
	printf("Enter the elements of the array : ");
	for(i=0;i<num;i++)
		scanf("%d",&arr[i]);
	for(i=0;i<num;i++)
		push(arr[i]);
	for(i=0;i<num;i++)
	{
		x=pop();
		arr[i]=x;
	}
	printf("the revesed array is : ");
	for(i=0;i<num;i++)
		printf("%d ",arr[i]);
}

void push(int a)
{
	top=top+1;
	S[top]=a;

}
int pop()
{

	return S[top--];
}
```

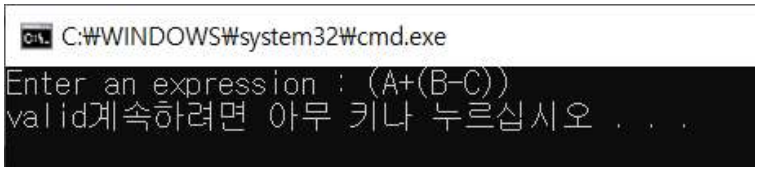
```
C:₩WINDOWS₩system32₩cmd.exe
Enter the number of elements in the array : 5
Enter the elements of the array : 1 2 3 4 5
the revesed array is : 5 4 3 2 1 계속하려면 아무 키나 누르십시오 . . .
```

5.

```
#include <stdio.h>
#include <string.h>
#define max 10
int top = -1;
int stack[max];
```

```
void push(char c);
char pop();
int main()
{
	char arr[max],temp;
	int i, flag=1;
	printf("Enter an expression : ");
	gets(arr);
	for(i=0;i<strlen(arr);i++)
	{
		if(arr[i]=='(' || arr[i]=='{' || arr[i]=='[')
			push(arr[i]);
		if(arr[i]==')' || arr[i]=='}' || arr[i]==']')
			if(top == -1)
				flag=0;
			else
			{
				temp=pop();
				if(arr[i]==')' && (temp=='{' || temp=='['))
					flag=0;
				if(arr[i]=='}' && (temp=='(' || temp=='['))
					flag=0;
				if(arr[i]==']' && (temp=='(' || temp=='{'))
					flag=0;
			}
	}
	if(top>-1)
		flag=0;
	if(flag==1)
		printf("valid");
	else
		printf("invalid");
}
void push(char c)
{
	if(top==max-1)
		printf("stack overflow");
	else
	{
		top=top+1;
		stack[top]=c;
	}
}
```

```
char pop()
{
        if(top==-1)
                printf("stack underflow");
        else
                return (stack[top--]);
}
```

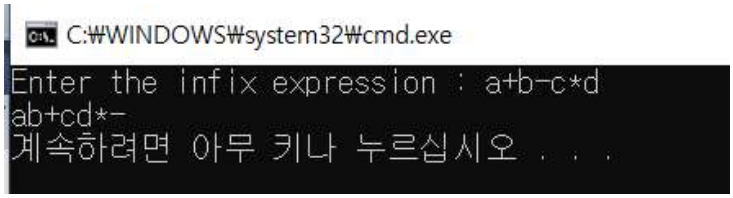


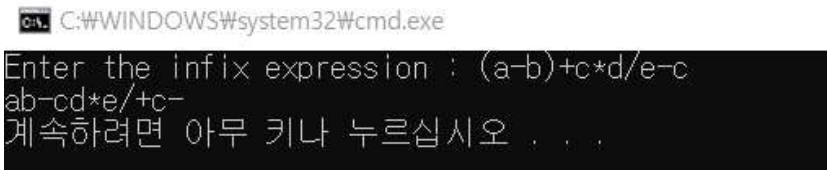
6.

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <stdlib.h>
# define max 500
char stack[max];
int top=-1;
void change(char input[], char output[]);
void push(char input[], char value);
char pop(char input[]);
int prior(char value);
int main()
{
        char input[500],output[500];
        int i;
        printf("Enter the infix expression : ");
        gets(input);
        strcpy(output, "");
        change(input,output);
        puts(output);

}


void change(char input[], char output[])
{
        int i=0,j=0;
        char temp;
        while(input[i]!='\0')
        {
```

```
		if(input[i]=='(')
		{
			push(stack,input[i]);
			i++;
		}
		else if(input[i]==')')
		{
			while((top!=-1) && (stack[top]!='('))
			{
				output[j]=pop(stack);
				j++;
			}
			if(top==-1)
			{
				printf("incorrect\n");
			}
			temp=pop(stack);
			i++;
		}
		else if(isdigit(input[i]) || isalpha(input[i]))
		{
			output[j]=input[i];
			i++;
			j++;
		}
		else if(input[i] == '+' || input[i] == '-' || input[i] == '*' || input[i] == '/' ||
input[i] == '%')
		{
			while((top!=-1) && (stack[top]!='(') &&
(prior(stack[top])>=prior(input[i])))
			{
				output[j]=pop(stack);
				j++;
			}
			push(stack,input[i]);
			i++;
		}
		else
		{
			printf("incorrect element in expression\n");
			exit(1);
		}
	}
```

```
        while((top!=-1) && (stack[top]!='('))
        {
                output[j]=pop(stack);
                j++;
        }
        output[j]='\0';
}

void push(char input[], char value)
{
        if(top==max-1)
                printf("Stack overflow\n");
        else
        {
                top++;
                stack[top]=value;
        }
}

char pop(char input[])
{
        char value=' ';//top==-1일때 초기화 해놓지 않으면 오류발생
        if(top==-1)
                printf("stack underflow");
        else
        {
                value=stack[top];
                top--;
        }
        return value;
}

int prior(char value)
{
        if(value=='/' || value=='*')
                return 3;
        if(value=='%')
                return 2;
        if(value=='+' || value=='-')
                return 1;
}
```





7.

```
#include <stdio.h>
#include <ctype.h>
# define max 100
double stack[max];
int top=-1;
double cal(char input[]);
void push(double input[], double value);
double pop(double input[]);
int main()
{
        char input[100];
        double value;
        printf("Enter any postfix expression :");
        gets(input);
        value=cal(input);
        printf("%.1lf",value);
}

double cal(char input[])
{
        int i=0;
        int value_1,value_2,value_3;
        while(input[i]!='\0')
        {
                if(isdigit(input[i]))
                        push(stack,(double)(input[i]-'0'));
                else
                {
                        value_1=pop(stack);
                        value_2=pop(stack);
```

```
                        if(input[i]=='+')
                                value_3=value_2+value_1;
                        if(input[i]=='-')
                                value_3=value_2-value_1;
                        if(input[i]=='/')
                                value_3=value_2/value_1;
                        if(input[i]=='*')
                                value_3=value_2*value_1;
                        if(input[i]=='%')
                                value_3=value_2%value_1;

                        push(stack,value_3);
                }

                i++;
        }
        return pop(stack);
}

void push(double input[], double value)
{
        if(top==max-1)
                printf("stack overflow");
        else
        {
                top++;
                stack[top]=value;
        }
}

double pop(double input[])
{
        double value=-1;
        if(top==-1)
                printf("Stack underflow");
        else
        {
                value=stack[top];
                top--;
        }
        return value;
}
```

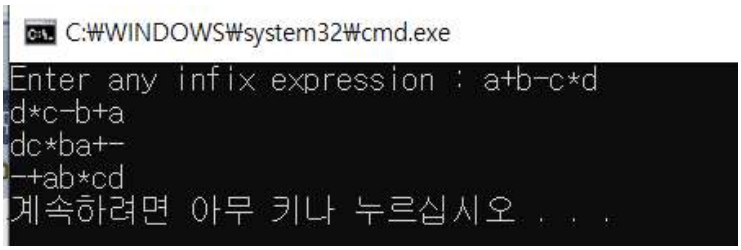
8.

```
#include <stdio.h>
#include <string.h>
#include <ctype.h>
#include <stdlib.h>
#define max 100
char stack[max];
char input[100],output[100], temp[100];
int top=-1;
void rev(char arr[]);
void change(char temp[], char output[]);
void push(char array[], char value);
char pop(char array[]);
int prior(char value);
int main()
{

        printf("Enter any infix expression : ");
        gets(input);
        rev(input);
        puts(temp);
        change(temp,output);
        puts(output);
        strcpy(temp,"");
        rev(output);
        puts(temp);
}

void rev(char arr[])
{
        int len, i=0, j=0;
        len=strlen(arr);
        j=len-1;
        while(j>=0)
        {
                if(arr[j]=='(')
                        temp[i]=')';
                else if(arr[j]==')')
                        temp[i]='(';
```

```
                else
                        temp[i]=arr[j];
                i++;
                j--;
        }
        temp[i]='\0';
}

void change(char temp[], char output[])
{
        int i=0,j=0;
        while(temp[i]!='\0')
        {
                if(temp[i]=='(')
                {
                        push(stack,temp[i]);
                        i++;
                }
                else if(temp[i]==')')
                {
                        while((top!=-1)&&(stack[top]!='('))
                        {
                                output[j]=pop(stack);
                                j++;
                        }
                        if(top==-1)
                        {
                                printf("incorrect");
                        }
                        pop(stack);
                        i++;
                }
                else if(isdigit(temp[i]) || isalpha(temp[i]))
                {
                        output[j]=temp[i];
                        i++;
                        j++;
                }
                else if(temp[i] == '+' || temp[i] == '-' || temp[i] == '*' || temp[i] == '/' ||
temp[i] == '%')
                {
                        while((top!=-1)         &&         (stack[top]!='(')         &&
(prior(stack[top])>prior(temp[i])))
```

```
                        {
                                output[j]=pop(stack);
                                j++;
                        }
                        push(stack,temp[i]);
                        i++;
                }
                else
                {
                        printf("incorrect element in expression");
                        exit(1);
                }
        }
        while(top!=-1 && stack[top]!='(')
                {
                        output[j]=pop(stack);
                        j++;
                }
        output[j]='\0';
}

void push(char array[], char value)
{
        if(top==max-1)
                printf("stack overflow");
        else
        {
                top++;
                stack[top]=value;
        }
}

char pop(char array[])
{
        char value= ' ';
        if(top==-1)
                printf("stack underflow");
        else
        {
                value=stack[top];
                top--;
        }
        return value;
```

```
}

int prior(char value)
{
	if(value=='*' || value=='/')
		return 3;
	if(value=='%')
		return 2;
	if(value=='+' || value=='-')
		return 1;
}
```



9.

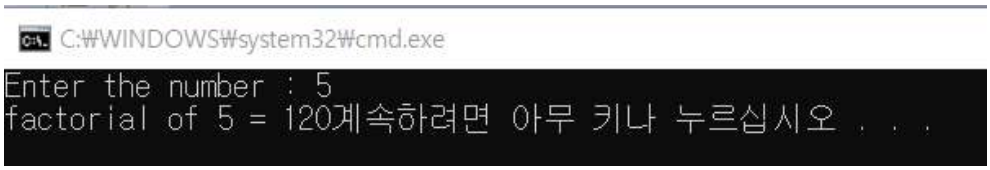
```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
# define max 100
double stk[max];
int top=-1;
char input[100];
double cal(char input[]);
void push(double input[], double value);
double pop();
int main()
{

	double value;
	printf("Enter any postfix expression :");
	gets(input);
	value=cal(input);
	printf("%.1lf",value);
}

double cal(char input[])
{
	int i;
	double value_1,value_2,value_3;
	i=strlen(input)-1;
```

```
	while(i>=0)
	{
		if(isdigit(input[i]))
			push(stk,(double)(input[i]-'0'));
		else
		{
			value_1=pop(stk);
			value_2=pop(stk);


			if(input[i]=='+')
				value_3=value_1+value_2;
			if(input[i]=='-')
				value_3=value_1-value_2;
			if(input[i]=='/')
				value_3=value_1/value_2;
			if(input[i]=='*')
				value_3=value_1*value_2;
			if(input[i]=='%')
				value_3=(int)value_1%(int)value_2;

			push(stk,value_3);
		}

		i--;
	}
	return pop(stk);
}

void push(double stk[], double value)
{
	if(top==max-1)
		printf("stk overflow");
	else
	{
		top++;
		stk[top]=value;
	}
}

double pop(double stk[])
{
```

```
        double value;
        if(top==-1)
                printf("stk underflow");
        else
        {
                value=stk[top];
                top--;
        }
        return value;
}
```

10.

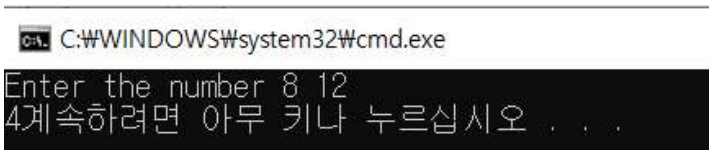
```
#include <stdio.h>

int main()
{
        int num,val;
        printf("Enter the number : ");
        scanf("%d",&num);
        val=fact(num);
        printf("factorial of %d = %d",num,val);

}
int fact(int n)
{
        if(n==1)
                return 1;
        else
                return (fact(n-1)*n);
}
```



11.

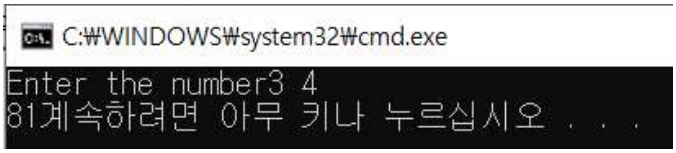
```
#include <stdio.h>

int main()
{
        int x,y;
```

```
	int result;
	printf("Enter the number");
	scanf("%d %d",&x,&y);
	result=cal(x,y);
	printf("%d",result);
}

int cal(int x, int y)
{
	int rem;
	rem=x%y;
	if(rem==0)
		return y;
	else
		return cal(y,rem);
}
```



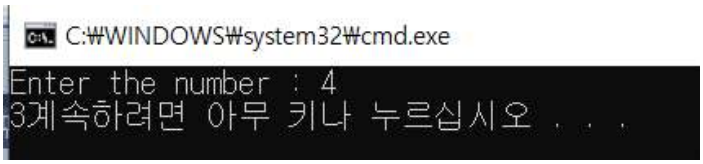
12.

```
#include <stdio.h>

int main()
{
	int x,y;
	int result;
	printf("Enter the number");
	scanf("%d %d",&x,&y);
	result=exp(x,y);
	printf("%d",result);
}

int exp(int x, int y)
{

	if(y==0)
		return 1;
	else
		return (x*exp(x,y-1));
}
```



13.

```
#include <stdio.h>

int main()
{
	int x;
	int result;
	printf("Enter the number : ");
	scanf("%d",&x);
	result=fb(x);
	printf("%d",result);
}

int fb(int x)
{

	if(x==0)
		return 0;
	else if(x==1)
		return 1;
	else
		return (fb(x-1)+fb(x-2));
}
```



Programming Exercises

1.

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

struct stack
{
        int data;
        struct stack *next;
};
struct stack *top =NULL;
struct stack *display(struct stack *top);
struct stack *push(struct stack *top, int val);
struct stack *pop(struct stack *top);
int main ()
{
        int option, val;

        do
        {       printf("\n\n");
                printf("1.push\n");
                printf("2.pop\n");
                printf("3.display\n");
                printf("4.exit");
                printf("Enter your option : ");
                scanf("%d", &option);
                switch(option)
                {
                case 1:
                        printf("Enter the number to be pushed on stack");
                        scanf("%d",&val);
                        top=push(top,val);
                        break;
                case 2:
                        top=pop(top);
                        break;
                case 3:
                        top=display(top);
                        break;
                }

        }while(option!=4);
```

```
}

struct stack *push(struct stack *top, int val)
{
        struct stack *new_node;
        new_node=(struct stack *)malloc(sizeof(struct stack));
        new_node->data=val;
        if(top==NULL)
        {
                new_node->next=NULL;
                top=new_node;
        }
        else
        {
                new_node->next=top;
                top=new_node;
        }
        return top;
}

struct stack *pop(struct stack *top)
{
        struct stack *ptr;
        ptr=top;
        if(top==NULL)
                printf("stack underflow");
        else
        {
                top=top->next;
                printf("the value being deleted is %d", ptr->data);
                free(ptr);
        }
        return top;
}

struct stack *display(struct stack *top)
{
        struct stack *ptr;
        ptr=top;
        if(top==NULL)
        {
                printf("stack is empty");
        }
```

```
        else
        {
                while(ptr!=NULL)
                {
                        printf("%d",ptr->data);
                        ptr=ptr->next;
                }
        }
        return top;
}
```

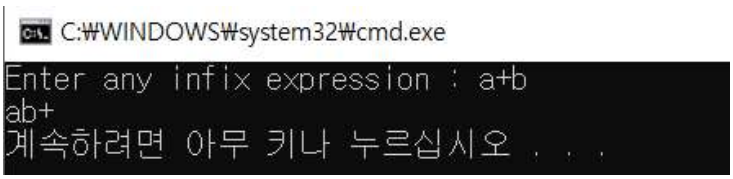
```
1.push
2.pop
3.display
4.exitEnter your option : 1
Enter the number to be pushed on stack1


1.push
2.pop
3.display
4.exitEnter your option : 1
Enter the number to be pushed on stack2


1.push
2.pop
3.display
4.exitEnter your option : 1
Enter the number to be pushed on stack3


1.push
2.pop
3.display
4.exitEnter your option : 3
321

1.push
2.pop
3.display
4.exitEnter your option : 4
계속하려면 아무 키나 누르십시오 . . .
```

2.

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <stdlib.h>
# define max 10
```

```
void push(char stack[], char val);
char pop(char stack[]);
void change(char infix[], char postfix[]);
char stack[max];
int top=-1;
int main()
{
	char infix[10], postfix[10];
	printf("Enter any infix expression : ");
	gets(infix);
	change(infix,postfix);
	puts(postfix);
}

void change(char infix[], char postfix[])
{
	int i=0;
	int j=0;
	while(infix[i]!='\0')
	{
		if(isdigit(infix[i]) || isalpha(infix[i]))
		{
			postfix[j]=infix[i];
			i++;
			j++;
		}
		else if(infix[i] == '+')
		{
			push(stack,infix[i]);
			i++;
		}
		else
		{
			printf("wrong");
			exit(1);
		}
	}
	while(top!=-1)
	{
		postfix[j]=pop(stack);
		j++;
	}
	postfix[j]='\0';
```

```
}

void push(char stack[], char val)
{
	if(top==max-1)
		printf("stack overflow");
	else
	{
		top++;
		stack[top]=val;
	}
}

char pop(char stack[])
{
	char val;
	if(top==-1)
		printf("stack underflow");
	else
	{
		val=stack[top];
		top--;
	}
	return val;
}
```

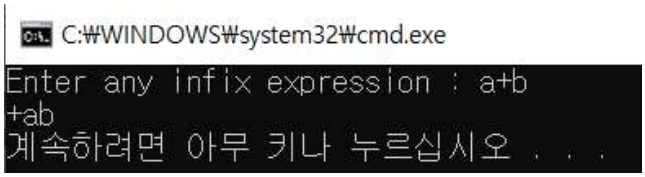


3.

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
#include <stdlib.h>
# define max 10
void rev(char infix[]);
void change(char infix[], char postfix[]);
void push(char stack[], char val);
char pop(char stack[]);
char stack[max];
char infix[10], postfix[10], temp[10];
int top=-1;
int main()
```

```
{
        printf("Enter any infix expression : ");
        gets(infix);
        rev(infix);
        change(temp,postfix);
        strcpy(temp,"");
        rev(postfix);
        puts(temp);
}
void rev(char infix[])
{
        int j,i=0;
        j=strlen(infix);
        j=j-1;
        while(j>=0)
        {
                temp[i]=infix[j];
                i++;
                j--;
        }temp[i]='\0';
}
void change(char infix[], char postfix[])
{
        int i=0;
        int j=0;
        while(infix[i]!='\0')
        {
                if(isdigit(infix[i]) || isalpha(infix[i]))
                {
                        postfix[j]=infix[i];
                        i++;
                        j++;
                }
                else if(infix[i] == '+')
                {
                        push(stack,infix[i]);
                        i++;
                }
                else
                {
                        printf("wrong");
                        exit(1);
                }
```

```
        }
        while(top!=-1)
        {
                postfix[j]=pop(stack);
                j++;
        }
        postfix[j]='\0';
}

void push(char stack[], char val)
{
        if(top==max-1)
                printf("stack overflow");
        else
        {
                top++;
                stack[top]=val;
        }
}

char pop(char stack[])
{
        char val;
        if(top==-1)
                printf("stack underflow");
        else
        {
                val=stack[top];
                top--;
        }
        return val;
}
```



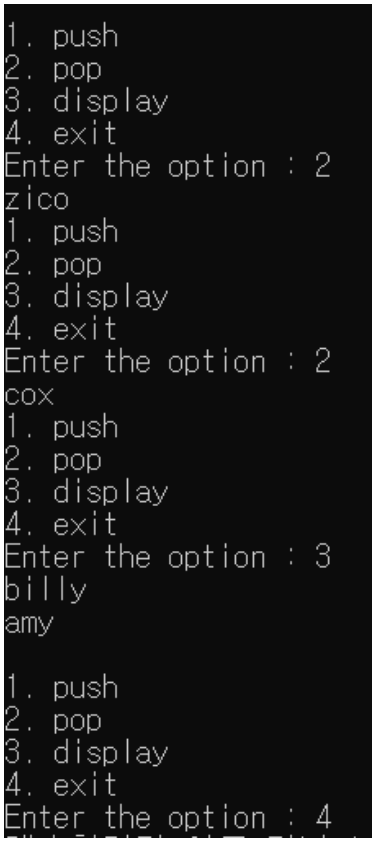
4.

```
#include <stdio.h>
#include <string.h>
#define max 5
void push(char name[5][10],char name_i[]);
void display(char name[5][10]);
char *pop(char *value ,char name[5][10]);
```

```
char name[max][10];
int top=-1;
int main()
{
	int i,option;
	char value[10];
	char name_i[10];
	do{
		printf("\n");
		printf("1. push\n");
		printf("2. pop\n");
		printf("3. display\n");
		printf("4. exit\n");

	printf("Enter the option : ");
	scanf("%d",&option);
	switch(option)
		{
	case 1:
		printf("Enter the name of student : ");
		scanf("%s",name_i);
		push(name,name_i);
		break;

	case 2:
		pop(value,name);
		printf("%s",value);
		break;

	case 3:
		display(name);
		break;
		}
	}while(option!=4);

}


void push(char name[5][10],char name_i[])
{
	if(top==max-1)
		printf("stack overflow");
	else
```

```
        {
                top++;
                strcpy(name[top],name_i);
        }
}

char *pop(char *value ,char name[5][10])
{
        char val[10];
        if(top==-1)
                printf("stack underflow");
        else
        {
                strcpy(value,name[top]);
                top--;
                return value;
        }
}

void display(char name[5][10])
{
        int i;
        if(top==-1)
                printf("stack is empty");
        else
        {
                for(i=top;i>=0;i--)
                printf("%s\n",name[i]);

        }
}
```

```
1. push
2. pop
3. display
4. exit
Enter the option : 1
Enter the name of student : amy

1. push
2. pop
3. display
4. exit
Enter the option : 1
Enter the name of student : billy

1. push
2. pop
3. display
4. exit
Enter the option : 1
Enter the name of student : cox

1. push
2. pop
3. display
4. exit
Enter the option : 1
Enter the name of student : zico

1. push
2. pop
3. display
4. exit
Enter the option : 3
zico
cox
billy
amy

1. push
2. pop
3. display
4. exit
Enter the option : 2
zico
```



5.

```
#include <stdio.h>
# define max1 5
# define max2 5
int stackA[max1];
int stackB[max2];
int top1=-1;
int top2=-1;
void pusha(int stack[],int val);
void pushb(int stack[],int val);
int main()
{
        int option,val,i;
        int flag=1;
        do{
        printf("\n");
        printf("1. push A stack\n");
        printf("2. push B stack\n");
        printf("3. compare the element\n");
        printf("4. exit\n");
        printf("Enter your option : ");
        scanf("%d",&option);
```

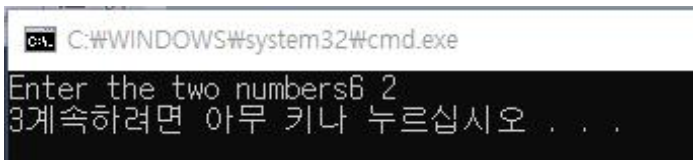
```
        switch(option)
        {
        case 1:
                printf("Enter the number : ");
                scanf("%d",&val);
                pusha(stackA,val);
                break;
        case 2:
                printf("Enter the number : ");
                scanf("%d",&val);
                pushb(stackB,val);
                break;
        case 3:
                for(i=top1;i>=0;i--)
                {
                        if(stackA[i]!=stackB[i])
                                flag=0;
                }
                if(flag==1)
                        printf("same");
                else
                        printf("not same");
                break;
        }

        }while(option!=4);

}

void pusha(int stack[],int val)
{
        if(top1==max1-1)
                printf("stack overflow");
        else
        {
                top1++;
                stack[top1]=val;
        }
}

void pushb(int stack[],int val)
{
        if(top2==max2-1)
```

```
                printf("stack  overflow");
        else
        {
                top2++;
                stack[top2]=val;
        }
}
```

```
1. push A stack
2. push B stack
3. compare the element
4. exit
Enter your option : 1
Enter the number : 1

1. push A stack
2. push B stack
3. compare the element
4. exit
Enter your option : 1
Enter the number : 2

1. push A stack
2. push B stack
3. compare the element
4. exit
Enter your option : 1
Enter the number : 3

1. push A stack
2. push B stack
3. compare the element
4. exit
Enter your option : 2
Enter the number : 1

1. push A stack
2. push B stack
3. compare the element
4. exit
Enter your option : 2
Enter the number : 2

1. push A stack
2. push B stack
3. compare the element
4. exit
Enter your option : 2
Enter the number : 4

1. push A stack
2. push B stack
3. compare the element
4. exit
Enter your option : 3
not same
1. push A stack
2. push B stack
3. compare the element
4. exit
Enter your option : 4
계속하려면 아무 키나 누르십시오 . . .
```

6.

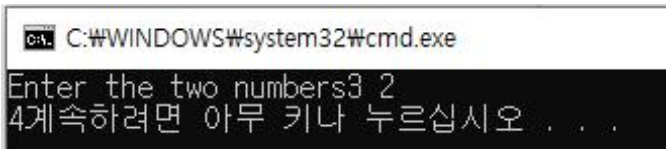
```
#include <stdio.h>

int main()
{
	int x,y;
	int res;
	printf("Enter 쿽 the two numbers");
	scanf("%d %d",&x,&y);
	res=F(x,y);
	printf("%d",res);
}

int F(int x, int y)
{
	if(x>=y)
		return (F(x-y,y)+1);
	else
		return 0;
}
```



7.

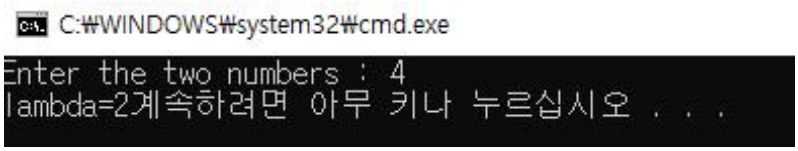
```
#include <stdio.h>

int main()
{
	int x,y;
	int res;
	printf("Enter the two numbers");
	scanf("%d %d",&x,&y);
	res=F(x,y);
	printf("%d",res);
}

int F(int x, int y)
{
	if(x==1)
		return 1;
	else
		return (F(x-1,y)+F(x-1,y-1));
```

```
}
```



8.

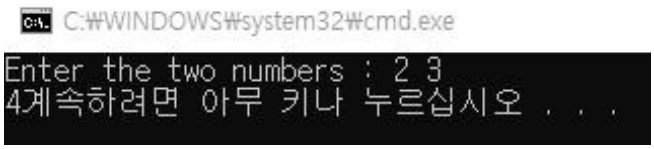
```
#include <stdio.h>

int main()
{
	int x;
	int res;
	printf("Enter the two numbers : ");
	scanf("%d",&x);
	res=lamda(x);
	printf("lambda=%d",res);
}

int lamda(int x)
{

	if(x==1)
		return 0;
	else if(x>1)
		return (lamda(x/2)+1);
	}
```



9.

```
#include <stdio.h>

int main()
{
	int x,y;
	int res;
	printf("Enter the two numbers : ");
	scanf("%d %d",&x,&y);
	res=F(x,y);
	printf("%d",res);
}
```

```
int F(int x, int y)
{

        if((x==0)||((x>=y)&&(y>=1)))
                return 1;
        else
                return (F(x-1,y)+F(x-1,y-1));
        }
```



10.

```
#include <stdio.h>

void reverse(char *s) ;

int main()
{
    char s[100];

    printf("입력 : ");
    gets(s);

    reverse(s);

}

void reverse(char *s)
{
    char c=*s;
        if(c!='\0')
                reverse(++s);
        printf("%c",c);
}
```