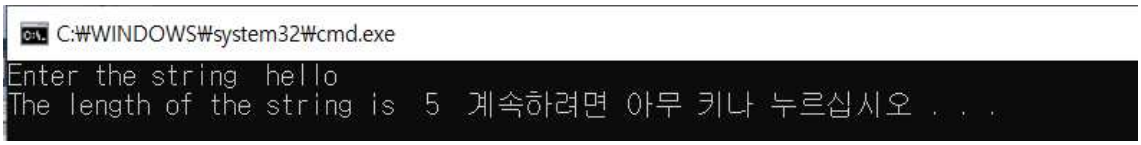
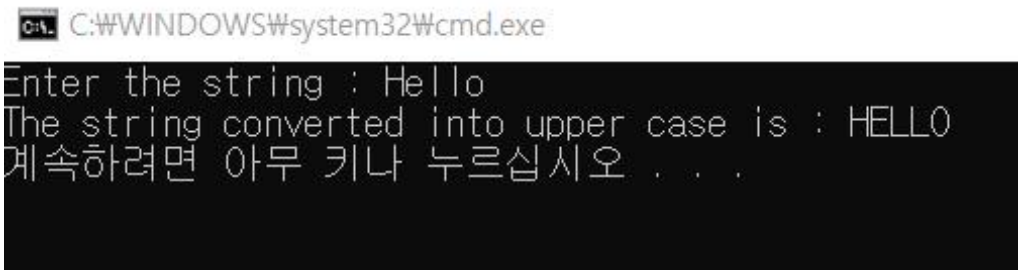
1.

```
#include <stdio.h>

int main()
{
	char a[100];
	int i=0;
	printf("Enter the string  ");
	gets(a);
	while(a[i]!=0)
	{
		i++;
	}

	printf("The length of the string is  %d  ",i);

}
```



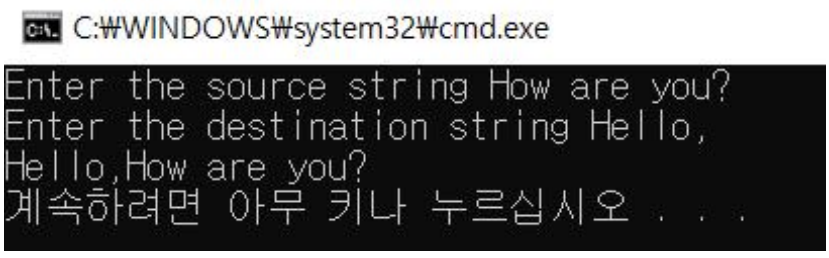
2.

```
#include <stdio.h>

int main()
{
	char a[100];
	int i=0;
	printf("Enter the string : ");
	gets(a);
	while(a[i]!=0)
	{
		if(a[i]>='a'&& a[i]<='z')
			a[i]=a[i]-32;
		else
			a[i]=a[i];
		i++;
	}
	printf("The string converted into upper case is : ");
	puts(a);
}
```



3.

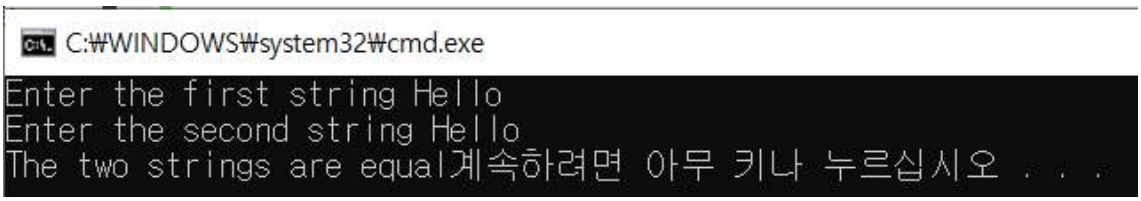
```
#include <stdio.h>

int main()
{
	char a[50];
	char b[50];
	int i=0,k=0;
	printf("Enter the source string ");
	gets(a);
	printf("Enter the destination string ");
	gets(b);
	while(b[i]!='\0')
	    i++;

	while(b[k]!='\0')
	{
		b[i]=a[k];
		i++;
		k++;

	}
	b[i]='\0';
	puts(b);


}
```



4.

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

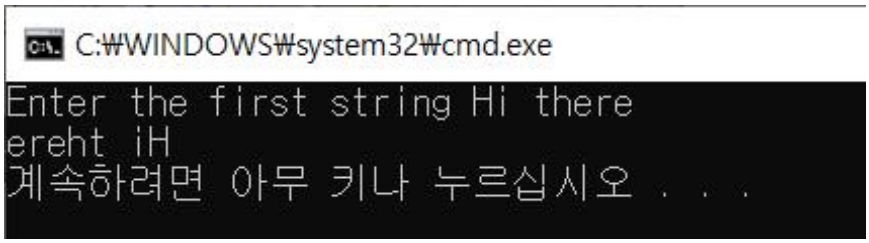
```
#include <string.h>
int main()
{
	char a[10],b[10];
	int i=0,j=0;
	int flag=0;
	printf("Enter the first string ");
	gets(a);
	printf("Enter the second string ");
	gets(b);
	if(strlen(a)==strlen(b))
		while(a[i]!='\0')
		{
			if(a[i]==b[i])
				flag=1;
			i++;
		}
	if(flag==1)
		printf("The two strings are equal");
	else
		printf("The two strings are not same");

}
```



5.

```
#include <stdio.h>
#include <string.h>
int main()
{
	char a[10],b[10];
	int i=0,j=0;
	int box;
	printf("Enter the first string ");
	gets(a);
	j=strlen(a)-1;
	while(i<j)
	{
		box=a[j];
		a[j]=a[i];
```

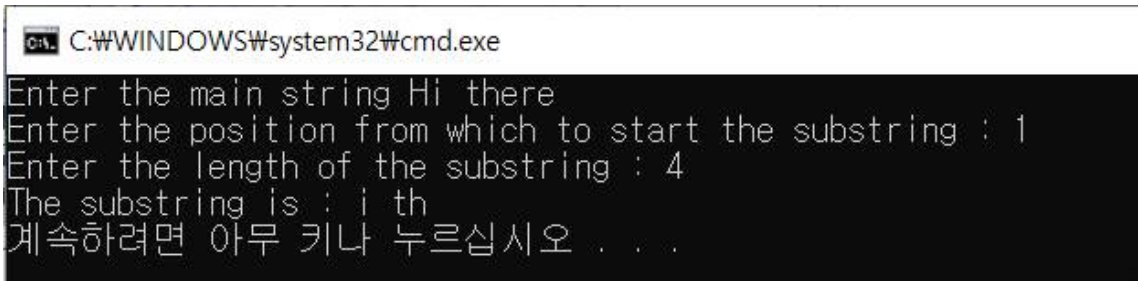
```
                a[i]=box;
                i++;
                j--;

        }

        puts(a);

}
```



6.

```
#include <stdio.h>
#include <string.h>
int main()
{
        char a[30],b[30];
        int pos,len;
        int k,j=0;
        printf("Enter the main string ");
        gets(a);
        printf("Enter the position from which to start the substring : ");
        scanf("%d",&pos);
        printf("Enter the length of the substring : ");
        scanf("%d",&len);
        k=pos;
        if(pos==0)
                len=len-1;
        while(k<=len)
        {
                        b[j]=a[k];
                        k++;
                        j++;
        }
        b[j]='\0';
        printf("The substring is : ");
        puts(b);
}
```

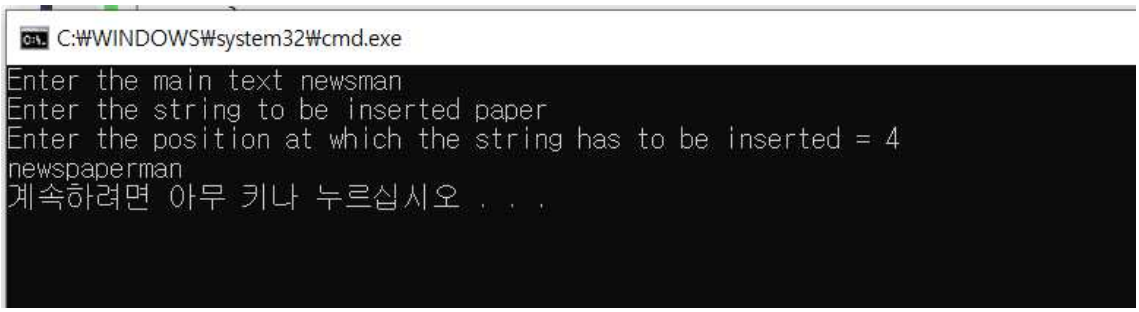
}



7.

```
#include <stdio.h>

int main()
{
	char a[20];
	char b[20];
	char c[40];
	int pos,i=0,j=0,q=0,flag=0;
	printf("Enter the main text ");
	gets(a);
	printf("Enter the string to be inserted ");
	gets(b);
	printf("Enter the position at which the string has to be inserted = ");
	scanf("%d",&pos);

	while(a[i]!='\0')
	{
		if(pos==i)
		{
			while(b[q]!='\0')
				{
					c[j]=b[q];
					q++;
					j++;
				}
			break;
		}
		c[j]=a[i];
		j++;
		i++;


	}
```

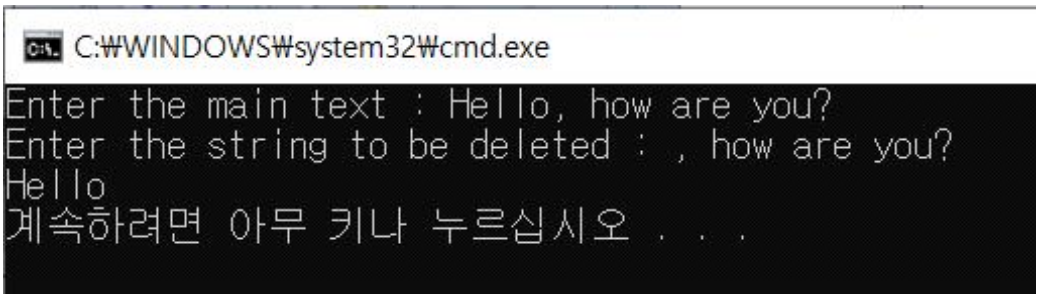
```
	while(c[i]!='\0')
	{
		c[j]=a[i];
		i++;
		j++;
	}
	c[j]='\0';
	puts(c);

}
```
/*책에 나와있는대로 하면 오류가 발생하는 이유는 if문에서 pos값을 검사(이때 i값은 4이다)하고 while문을 돌린뒤에 나온뒤 i값을 증가 시켜버리면
 i=4 즉 a행렬에 들어있는 5번째 값을 넣어주지 못한채 i=5로 넘어가서 6번째값부터 들어가게되기 때문에 완벽한 문장이 나오지 않게된다.*/



8.

```
#include <stdio.h>

int main()
{
	char main[200],del[20],new_text[200];
	int i=0,j,k,x=0,q=0;
	printf("Enter the main text : ");
	gets(main);
	printf("Enter the string to be deleted : ");
	gets(del);
	while(main[i]!='\0')
	{		j=0,k=i;/*j값을 0으로 초기화하는 것은 들어오는 i값에 대하여 처음부터 검사하기 위함이며 i의 값을 k에 넣는 이유는 두번째 while문안에서 검사가 이뤄질때 i값을 그대로사용하면 i값이 변화하면서 후에 들어갈 문자를 뛰어넘게 된다.*/
		while(main[k]==del[j]&&del[j]!='\0')
		{
			j++;
			k++;
		}/*k값을 계속 올리면서 del문자열과 같은 패턴을 찾아낸다*/
		if(del[j]=='\0')
```

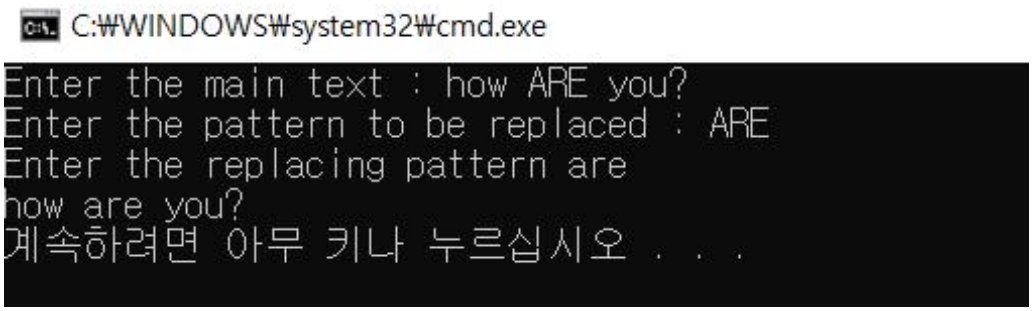
```
                        q=k;/*del문자열의 공백값을 만나면 main의 삭제되지않은 문자열의 인덱스값을 q에 넣는다.*/

                new_text[x]=main[q];
                x++;//new_text의 인덱스값
                q++;//k를 받은 q값
                i++;
        }
                new_text[x]='\0';
                puts(new_text);
}
```



9.

```
#include <stdio.h>

int main()
{
        char main[200],pattern[20],replace[20],new_text[200];
        int i=0,j,k,x=0,q=0,n=0;
        printf("Enter the main text : ");
        gets(main);
        printf("Enter the pattern to be replaced : ");
        gets(pattern);
        printf("Enter the replacing pattern ");
        gets(replace);
        while(main[i]!='\0')
        {               j=0,k=i;
                while(main[k]==pattern[j]&&pattern[j]!='\0')
                {
                        j++;
                        k++;
                }
                if(pattern[j]=='\0')
                {
                        q=k;
                        while(replace[n]!='\0')
                        {
```

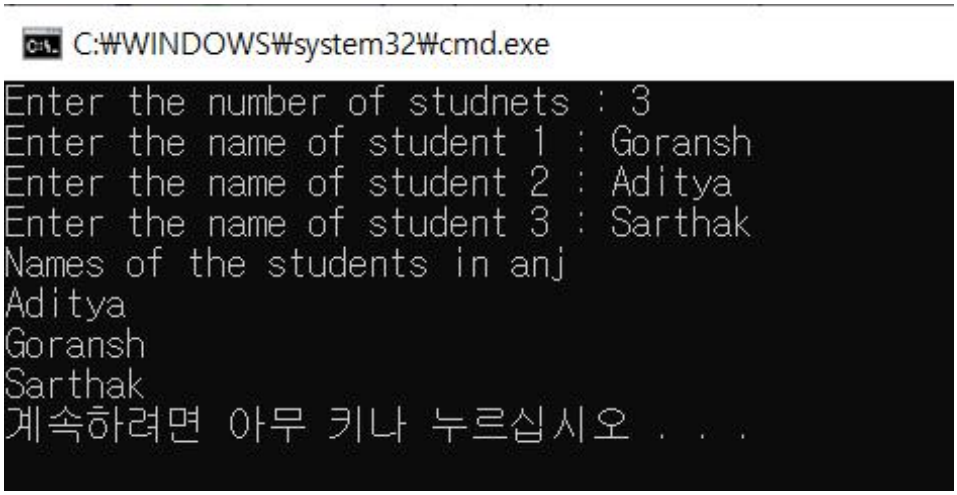
```
                                        new_text[x]=replace[n];
                                        x++;
                                        n++;
                                }
                        }
                        new_text[x]=main[q];
                        x++;
                        i++;
                        q++;
                }
                        new_text[x]='\0';
                        puts(new_text);
}
```



10.

```
#include <stdio.h>
#include <string.h>
void bubble_sort(char name[5][10],int n);
int main()
{
        char name[5][10];
        int i,n,j=0;
        printf("Enter the number of studnets : ");
        scanf("%d",&n);
        for(i=0;i<n;i++)
        {
                printf("Enter the name of student %d : ",i+1);
                scanf("%s",name[i]);
        }


        bubble_sort(name,n);
        printf("Names of the students in anj\n");
        for(i=0;i<n;i++)
                puts(name[i]);
}
```

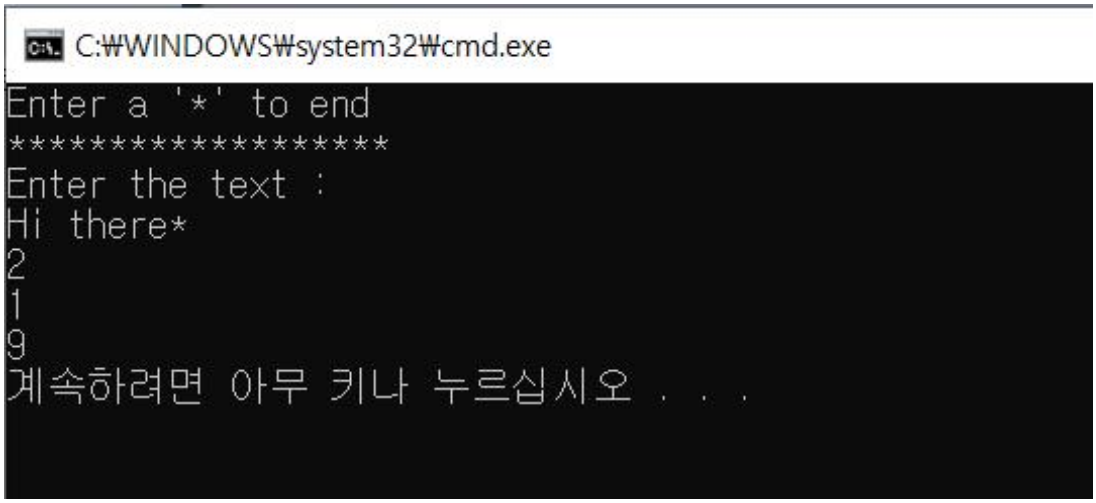
```
void bubble_sort(char name[5][10],int n)
{
	int i,j;
	char box[10];
	for(i=0;i<n;i++)
	{
		for(j=0;j<n-i-1;j++)
		{
			if(strcmp(name[j],name[j+1])>0)
			{
				strcpy(box,name[j]);
				strcpy(name[j],name[j+1]);
				strcpy(name[j+1],box);
			}
		}
	}
}
```



11.

```
#include <stdio.h>

int main()
{

	char text[30];
	int i=0,j=0;
	int a=1;//줄세기
	int b=1;//단어세기
	int c=0;//글자세기
	printf("Enter a '*' to end\n");
```

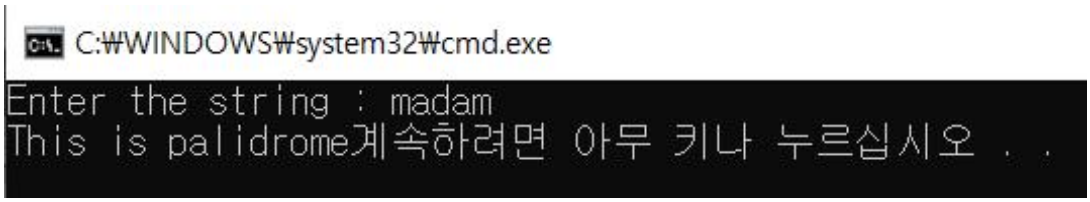
```
	printf("******************\n");
	printf("Enter the text : \n");

	while(text[i]!='\0')
	{
		i++;
		scanf("%c",&text[i]);
		if(text[i]=='*')
			break;

	}
	text[i]='\0';


	while(text[j]!='\0')
	{
		if(text[j]=='\n')
			a++;
		if(text[j]==' '&&text[j+1]!=' ')
			b++;

		c++;
		j++;

	}
	c=c-a+1;
	printf("%d\n",b);
	printf("%d\n",a);
	printf("%d\n",c);
}
```



12.

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

int main()
{
        char a[20];
        int len=0;
        int i=0,r=0,j=0;
        int flag=0;
        printf("Enter the string : ");
        gets(a);
        while(a[i]!='\0')
                {
                        len++;
                        i++;
            }


        if(len%2!=0)
                {
                        j=len-1;
                while(r<j)
                {
                        if(a[r]==a[j])
                        {

                                r++;
                                j--;
                                if(j==(len/2))
                                        flag=1;
```

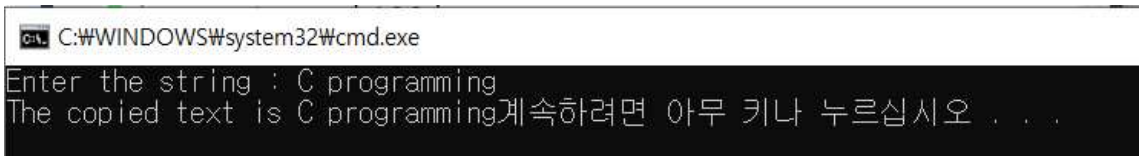
```
                        }
                        else
                                break;
                }
        }
         if(flag==1)
                 printf("This is palidrome");
         else
                 printf("This is not palidrome");

}
```



13.

```
#include <stdio.h>

int main()
{
        char a[50];
        char b[50];
        char *p1;
        char *p2;
        int i=0;
        int count=0;
        p1=a;
        p2=b;
        printf("Enter the string : ");
        gets(a);
        while(*p1!='\0')
        {
                *p2=*p1;

                p1++;
                p2++;
                count++;
        }
        *p2='\0';

        p2=p2-count;
```

```
        printf("The copied text is ");
        while(*p2!='\0')
        {
                printf("%c",*p2);
                p2++;
        }


}
```



14.

```
#include <stdio.h>

int main()
{
        char a[50], b[50], c[50];
        char *p1,*p2,*p3;
        int count=0;
        p1=a,p2=b,p3=c;


        printf("Enter the first string\n");
        gets(a);
        printf("Enter the second string\n");
        gets(b);
        while(*p1!='\0')
        {
                *p3=*p1;
                p1++;
                p3++;
                count++;
        }
        while(*p2!='\0')
        {
                *p3=*p2;
                 p3++;
            p2++;
                 count++;
        }
        *p3='\0';
```

```
        p3=p3-count;

        printf("The concatennated text is ");
        while(*p3!='\0')
        {
                printf("%c",*p3);
                p3++;
        }


}
```