

OUTPUT gcc 1.c

./a.out MENU -----

1. Create
2. Display
3. insert
4. delete
5. exit

enter your choice : 1

enter the no. of elements : 3

enter the elements into the array

1
2
3

----- MENU -----

1. Create
2. Display
3. insert
4. delete
5. exit

enter your choice : 2

the array elements are : 1

2
3

----- MENU -----

1. Create
2. display
3. insert
4. delete
5. exit

enter your choice : 3

enter the position for new element : 3

enter the element to be inserted : 5

PROGRAM-1

Design and develop and Implement a menu driven program in C for array operation.

- create an array of n integer elements.
- display of array elements with suitable headings.
- inserting an element at a given valid position.
- deletion of an element at a given valid position.
- exit.

support the program with functions for each of above operations.

```
#include<stdio.h>
#include<conio.h>
int a[10], n, elem, i, pos;
void create()
{
    printf("Enter the no. of elements");
    scanf("%d", &n);
    printf("Enter the elements into the array\n");
    for(i=0; i<n; i++)
        scanf("%d", &a[i]);
}
void display()
{
    printf("The array elements are!");
    for(i=0; i<n; i++)
        printf("%d\n", a[i]);
}
```

----- MENU -----

1. create
2. display
3. insert.
4. delete
5. exit.

enter your choice : 2

The array elements are : 1

2
3
5

----- MENU -----

1. create
2. display
3. insert.
4. delete
5. exit.

enter your choice : 4

enter the position of the elements to be deleted: 2

deleted element is 3


```
void insert()
```

```
{
```

```
printf("\n Enter the position for the new element:");
scanf("%d", &pos);
```

```
if(pos > n)
```

```
{
```

```
printf("Invalid position");
```

```
return;
```

```
}
```

```
printf("\n Enter the elements to be inserted:");
scanf("%d", &elem);
```

```
for(i = n-1; i >= pos; i--)
```

```
a[i+1] = a[i];
```

```
a[pos] = elem;
```

```
n = n+1;
```

```
}
```

```
void del()
```

```
{
```

```
printf("Enter the position of the elements to be deleted:");
scanf("%d", &pos);
```

```
if(pos > n-1)
```

```
{
```

```
printf("Invalid position");
```

```
return;
```

```
}
```

```
elem = a[pos];
```

```
for(i = pos; i < n-1; i++)
```

```
a[i] = a[i+1];
```

```
n = n-1;
```

```

printf("deleted element is %d", elem);
}
void main()
{
    int ch;
    do
    {
        printf("\n --- Menu --- \n");
        printf("1.create\n 2.Display\n 3.insert\n 4.Delete\n 5.exit\n");
        printf("Enter your choice:");
        scanf("%d", &ch);
        switch(ch)
        {
            case 1 : create(); break;
            case 2 : display(); break;
            case 3 : del(); break;
            case 4 : insert(); break;
            case 5 : exit(0); break;
            default : printf("Invalid choice \n");
        }
    } while (ch != 5);
}

```

10

30/3/17

OUTPUT: gcc 2.c

./a.out

enter the main string: Test

enter the pattern string: re

enter the replace string: te

resultant string is : rest

D	D	M	M	Y	Y	Y	Y
2	3	0	8	2	0	1	7

PROGRAM - 2

Design, develop and implement a menu driven program in C for the following operations on string.

- Read a main string (STR), pattern string (PAT) and to a replace string (REP).
- Perform pattern matching operation. Find and replace all occurrences of PAT in STR with REP if PAT exists in STR.

Support the program with function for each of the above operations don't use builtin function.

```
#include <stdio.h>
#include <string.h>
char str[100], pat[100], rep[50], ans[100];
void stringmatch()
{
    i = m = c = j = 0;
    while (str[c] != '\0')
    {
        if (str[m] == pat[i])
        {
            i++;
            m++;
            if (pat[i] == '\0')
            {
                flag = 1;
                for (k = 0; rep[k] != '\0'; k++, j++)
                    ans[j] = rep[k];
                i = 0;
                c = m;
            }
        }
    }
}
```

```

    }
else
{
    ans[i] = str[c];
    j++;
    c++;
    m=c;
    i=0;
}
}
ans[j] = "\0";
}

void main()
{
    printf("\n enter a main string\n");
    gets(str);
    printf("\n Enter a pattern string\n");
    gets(pat);
    printf("Enter a replace string\n");
    gets(rep);
    Stringmatch();
    if(flag==1)
        printf("\n the resultant string is\n%s",ans);
    else
        printf("\n Pattern string NOT Found\n");
    getch();
}

```


OUTPUT gcc 3.c
./a.out

----- MAIN MENU -----

1. PUSH
2. POP
3. PALLINDROME
4. EXIT

enter your choice : 1

enter a element to be pushed 2

the stack elements are: 2

----- MAIN MENU -----

1. PUSH
2. POP
3. PALLINDROME
4. EXIT

enter your choice : 1

enter your elements to be pushed 3

The stack elements are: 3
2

----- MAIN MENU -----

1. PUSH
2. POP
3. PALLINDROME
4. EXIT

enter your choice : 1

enter a element to be pushed: 3

the stack contents are: 3
3
2

PROGRAM-03

Design and develop, implement a menu driven program for the following operations on string STACK.

a) push() b) pop() c) palindrome() (e) display() (f) exit()

```
#define MAX 4
```

```
int stack[MAX], item;
```

```
int ch, top=-1, count=0, status=0;
```

```
void push(int item)
```

```
{
```

```
    if (top == (MAX-1))
```

```
        printf("\n\n Stack is overflow");
```

```
    else
```

```
    {
```

```
        stack[++top] = item;
```

```
        status++;
```

```
    }
```

```
}
```

```
int pop()
```

```
{
```

```
    int del;
```

```
    if (top == -1)
```

```
        printf("\n stack is underflow\n");
```

```
    else
```

```
    {
```

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

```
        status--;
```

```
        printf("\n popped element is %d", del);
```

```
    }
```

```
    return del;
```

```
}
```

----- MAIN MENU -----

1. PUSH

2. POP

3. PALLINDROME

4. EXIT

enter your choice : 1

enter a element to be pushed : 2

The stack contents are!

2
3
3
2

----- MAIN MENU -----

1. PUSH

2. POP

3. PALLINDROME

4. EXIT

enter your choice 3

stack contents are pallindrome.

D	D	M	M	Y	Y	Y	Y
3	0	0	8	2	0	1	7

```
void pallindrome( )
```

```
{
```

```
    int i, temp, j;
```

```
    temp = status;
```

```
    for (i = 0, j = top; i < temp; i++, j--)
```

```
    {
```

```
        if (stack[i] == stack[j])
```

```
            count++;
```

```
    }
```

```
    if (temp == count)
```

```
        printf("In stack contents are pallindrome");
```

```
    else
```

```
        printf("In stack contents are not pallindrome");
```

```
}
```

```
void display( )
```

```
{
```

```
    int i;
```

```
    printf("In the stack contents are:");
```

```
    if (top == -1)
```

```
        printf("In stack is empty");
```

```
    else
```

```
    {
```

```
        for (i = top; i >= 0; i--)
```

```
            printf("In %d ", stack[i]);
```

```
        }
```

```
    }
```

D	D	M	M	Y	Y	Y	Y
3	0	0	8	2	0	1	7

```
void main()
```

```
{
```

```
do
```

```
{
```

```
printf("\n ----- MAIN MENU ----- \n");
```

```
printf("\n 1. PUSH \n 2. POP \n 3. PALLINDROME \n 4. Exit");
```

```
printf("\n Enter your choice: ");
```

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

```
switch(ch)
```

```
{
```

```
case 1 : printf("\n Enter an element to be pushed");
```

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

```
push(item); display(); break;
```

```
case 2 : item = pop(); display(); break;
```

```
case 3 : pallindrome(); break;
```

```
case 4 : exit(0); break;
```

```
default: printf("\n END OF EXECUTION");
```

```
}
```

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

```
}
```

10

13/9