

21/12/23

LAB : 02

1) Swapping using pointers :

```
#include<stdio.h>
void swapref(int *a, int *b);
void main()
{
    int a = 10;
    int b = 15;
    printf("the values of a and b before swapping
are: \n %d\n %d", a, b);
    swapref(&a, &b);
    printf("swapping of two values after call by
reference \n %d\n %d", a, b);
}
```

```
void swapref(int *a, int *b)
```

```
{
    *int temp;
    temp = *a;
    *a = *b;
    *b = temp;
```

```
printf("the values of a and b after
swapping: \n %d\n %d", a, b);
```

Output:

the values of a and b before swapping:

10

15

15

10.

the values of a and b after swapping:

2) Dynamic memory:

```
#include<stdio.h>
void MallocEx(int);
void CallocEx(int);
void main()
{
    int n;
    printf("Enter the value of n : \n");
    scanf("%d", &n);
    MallocEx(n);
    CallocEx(n);
}
```

```
void MallocEx(int n)
```

```

int *ptr;
int i;
int arr[n];
ptr=(int *)malloc(n * sizeof(int));
for(i=0; i<n; i++)
    ptr[i]=i+1;
printf("Malloc dynamic memory allocation");
printf("the elements of the array are: \n");
for(i=0; i<n; i++)
    printf("%d", ptr[i]);
printf("\n");
free(ptr);
```

```
void CallocEx(int n)
```

```

int *ptr;
int i;
```

```

int arr[n];
ptr = (int *)calloc(n, sizeof(int));
for(i=0; i<n; i++)
    printf("%d", ptr[i]);
printf("calloc dynamic memory allocation\n");
printf("the elements of the array are:\n");
n=15
ptr = (int *)realloc(ptr, n * sizeof(int));
for(i=10; i<n; i++)
    ptr[i] = i+1;
for(i=0; i<n; i++)
    printf("%d", ptr[i]);
free(ptr);
}

```

Output:

Enter the value of n:

7

Malloc dynamic memory allocation
 the elements of the array are:

1 2 3 4 5 6 7

Calloc dynamic memory allocation
 the elements of the array are:

1 2 3 4 5 6 7

Realloc dynamic memory allocation

the elements of the array are:

1 2 3 4 5 6 7 0 5 8 9 2 0 2 5 9 3 2 5 2 7 9 4 1 2 1 3 1 4 1 5.

3) Stack implementation:

```
#include <stdio.h>
#include <stdlib.h>
#define SIZE = 5
int top = -1;
int inp_array [SIZE];
void push()
{
    int x;
    if (top == SIZE-1)
        printf("overflow\n");
    else
    {
        printf("Enter the element to be added
in the stack:\n");
        scanf(".d", &x);
        top = top + 1;
        inp_array [top] = x;
    }
}

void pop()
{
    if (top == -1)
        printf("Underflow\n");
    else
    {
        printf("Enter the element to be remo-
ved:\n");
        scanf("d", &x);
        printf("popped element is .d\n", inp_array
[top]);
    }
}
```

```
        } top = top - 1;  
    }  
void show()  
{  
    if (top == -1)  
        printf("Underflow\n");  
    else  
    {  
        printf("Elements in the stack are:\n");  
        for (int i = top; i >= 0; --i)  
            printf("%d\n", inp_array[i]);  
    }  
}
```

Output:

Operations on the stack:

1. push the element
2. pop the element
3. show
4. End

Enter the choice:

3

Underflow

Operations on the stack:

1. push the element
2. pop the element
3. show
4. End

Enter the choice

1

Enter the element to be added in the stack

5

operations on the stack:

1. push the element
2. pop the element
3. show

4. End

Enter the choice:

1

Enter the element to be added in the stack:

3

operations on the stack:

1. push the element
2. pop the element

3. show

4. End

Enter the choice:

2

Enter the element to be removed:

3

popped element : 3.

Ex. 1