Lab-2 :-

21/12/23 Date __/__/__
Page____

① Swap :-

```
#include <stdio.h>
 void swap(int *, int *);
 void main()

{
    int a, b;
    printf ("enter the value of a and b : \n");
    scanf  (" %d %d", &a, &b);
    swap (&a, &b);
}
    void swap (int * p, int * q)
{
    int temp =*p;
    * p = *q;
    *q = temp;
    printf (" the swap values are : %d %d", *p, *q);
```

Output :-

```
Enter the value of a and b:
 9
 6
the swap values are : 6 9
```

② Dynamic allocation :-

```
#include <stdio.h>
void Malloc (int);
void Calloc (int);
void Realloc (int);
void main ()
{
    int * ptr ;
    int n, i ;
    printf ("enter the value of n: \n");
    scanf (" %d ", &n);
    Malloc (n);
    Calloc (n);
    Realloc (n);
}
void Malloc (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");
printf ("the elements of array are \n");
  for (i =0; i<n; i++) {
           printf ("%d; ", ptr [i]);
     }
   printf ("\n")
   free (ptr);
}
```

Date

Page

```
void calloc (int n)
{
    int * ptr;
    int i;
    int arr [n];
    ptr =(int*) calloc (n, size of (int));
    for (i=0; i<n; i++)
    {
        ptr [i] = i+1;
    }
    printf ("\n");
     free (ptr);
}
Void Realloc(int n)
{
    int *ptr;
    int i;
    it arr [n];
    n=10;
    ptr = (int *) realloc (ptr, n* size of (int));
    for (i=5; i<n; i++)
     {
        ptr (i) = i+1;
     }
    ~~for(i=0;i<n;~~
     for (i=0; i<n; i++)
     {
        printf (" %d", ptr[i]);
     }
     free (ptr);
}
```

Output :-

```
enter the value of n:
 5
Malloc
the elements of array are:
 1 2 3 4 5
Calloc
 the elements of array are:
 1 2 3 4 5 0
Realloc
the elements of array are:
1 2 3 4 5 6 7 8 9 10
```

③ Stack :-

```
# include <stdio.h>
# include <stdlib.h>
# define SIZE 4
int top = -1;
 int inp_array [SIZE];
void push ();
void pop ();
void show ();

void main ()
 {
       int ch;
       while (1)
    {
       printf ("operation on the stack : \n");
  printf ("1. push the element \n 2. pop the element \n 3.
       show \n 4. End \n");
```

NP
21/12/2023

```
   printf("Enter the choice : \n");
   Scanf("%d", &ch);

 Switch (ch)
  {
    Case 1:
       push();
        break;
    Case 2:
       pop();
        break();
     Case3;
       Show();
       break;
     Case 4:
       exit(0);
   default:
       printf("Invalid Choice\n");
    }
  }
}
 void push ()
 {
     int x;
      if (top == SIZE - 1)
      {
         printf("Overflow\n");
      }
    else
     {
       printf("Enter the element to be added
        scanf("%d", &x);
    top = top + 1;
    inp_array [top] = x;
 }
}

   if top (== -1)
    {
      printf ("Underflow\n");
    }
   else
    {
      printf ("popped element : %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 :-

```
Enter the choiG :
1
Enter the element to be added :
5
Operations on the stack :
1. push
2.  PoP
3. Show
4. End
Enter the choiG :
1
Enter the element to be added:
6
Operation on stack:
1. push
2. pop
3. Show
4. end
Enter the choiG:
3
Elements in stack are:
6
5
```