21-12-23

Lab - 2

1) Swapping using pointers

```
#include <stdio.h>
void swap (int *, int *);
void main ()
{
    int a, b;
    printf ("enter two numbers a and b : ");
    scanf ("%d%d", &a, &b);
    printf ("value of a and b before swapping : %d %d \n", a,
    swap (&a, &b);
    printf ("values in main function after swapping : %d %d \n", a, b)
}

void swap (int *p, int *q)
{
    int temp;
    temp = *p;
    *p = *q;
    *q = temp;
    printf ("value of a and b after swapping : %d %d \n", *p,
}
```

Output:

enter two numbers a and b: 4 5
value of a and b before swapping: 4 5
values of a and b after swapping: 5 4
values of in main function after swapping: 5 4

2) Dynamic memory allocation

```
#include <stdio.h>
void Malloc (int);
void Calloc (int);

void main()
{
    int n;
    printf ("enter the number of elements n: \n");
    scanf ("%d", &n);
    Malloc (n);
    Calloc (n);
}

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

void Calloc (int n)
{
    int *ptr;
    int i;
    ptr = (int *) calloc (n, sizeof (int));
```

```
for(i=0;i<n;i++)
    ptr[i] = i+1;
printf("calloc memory allocation: \n");
printf("the elements are: \n");
for(i=0; i<n; i++)
    printf("%d ", ptr[i]);
n=10;
ptr = (int*)realloc(ptr, n*sizeof(int));
for(i=5; i<n; i++)
    ptr[i] = i+1;
printf("\n realloc memory allocation ");
printf("\n the elements are: \n");
for(i=0; i<n; i++)
    printf("%d", prt[i]);
free(ptr);
}
```

Output:

enter the number of elements n: 5
malloc memory allocation
the elements are:
1 2 3 4 5
calloc memory allocation
the elements are:
1 2 3 4 5
realloc memory allocation
the elements are:
1 2 3 4 5 6 7 8 9 10

3) Stack Implementation

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

void main()
{
    int ch;
    while (1)
    {
        printf("Operations on the stack: \n");
        printf("1. Push the element 2. Pop the element 3. Show 4. End \n");
        printf("Enter the choice: \n");
        scanf("%d", &ch);
        switch(ch)
        {
            case 1: push();
                    break;
            case 2: pop();
                    break;
            case 3: show();
                    break;
            case 4: exit(0);
            default: printf("Invalid choice");
        }
    }
}
```

```
void push ()
{
    int x;
    if (top == SIZE - 1)
        printf ("Overflow \n");
    else
    {
        printf ("Enter the element to be added: \n");
        scanf ("%d", &x);
        top = top + 1;
        a[top] = x;
    }
}
void pop ()
{
    if (top == -1)
        printf ("Underflow \n");
    else
    {
        printf ("Popped element : %d \n", a[top]);
        top = top - 1;
    }
}
void show ()
{
    if (top == -1)
        printf ("Underflow \n");
    else
    {
        printf ("Elements is the stack are: \n");
        for (int i = top ; i >= 0; i--)
            printf ("%d \n", a[i]);
    }
}
```

Output:

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: 3
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: 2
Operations on the stack:
1. Push the element 2. Pop the element 3. show 4. End
Enter the choice: 2
Popped element: 2
Operations on the stack:
1. Push the element 2. Pop the element 3. show 4. End
Enter the choice: 3
Elements is the stack are: 3
Operations on the stack:
1. Push the element 2. Pop the element 3. show 4. End
Enter the choice: 4

Sol:
21/12/23