Lab - 2 21/12/23

1) Swapping Using Pointers

```
#include <stdio.h>
#include <stdlib.h>
void swap (int *a, int *b)
    int temp = *a;
      *a = *b;
      *b = temp;
}
int main ()
{
    int a = 3, b = 4;
    printf (" Values of a and b are %d %d ", a,b);
     swap (&a, &b);
    printf (" Values after swapping is %d %d ', a, b);
     return 0;
}
```

Output - Values of a and b are 3 4
Values after swapping is 4 3

2) Memory Allocation (malloc, calloc, realloc)

```
#include <stdio.h>
#include <stdlib.h>
int size = 3
int main() {
  int* arr = malloc(size * sizeof(int));
  int* arr2 = calloc(size, sizeof(int));
  printf("arr and arr2 can store %d integers", size);
  realloc(arr, 2* size * sizeof(int));
  printf("\n Can now store %d integers in arr and %d in arr2\n",
  2* size, size);
  for(int i=0; i<size; i++){
      arr2[i] = 1;
  }
  for(int i=0; i<2* size; i++){
      arr[i] = 0;
  }
  printf("arr: ");
  for(int i=0; i<2* size; i++){
    printf(" %d ", arr[i]);
  }
  printf("\n arr2: ");
  for(int i=0; i<size; i++){
     printf(" %d ", arr[i]);
  }
   free(arr); free(arr2);
   return 0;
}
```

Output -

arr and arr2 can store 2 integers
can now store 6 integers in arr and 3 integers in arr2
arr: 0 0 0 0 0 0
arr2: 1 1 1

3) Stack Implementation -

```
#include <stdio.h>
#include <stdlib.h>
#define size 10
int top = -1;
int stack[size];
void push(int a);
int pop();
void display();
int main(){
  printf("1. Push In 2. Pop In 3. Display stack In 4. Exit In Enter choice:");
  int choice;
    int a;
    scanf("%d", &choice);
    while(choice != 4){
      switch(choice){
    case 1:
      printf("Enter integer to be pushed");
       scanf("%d", &a);
       push(a);
       break;
  case 2:
      a = pop();
       printf("Integer popped = %d In", a);
        break;
  case 3:
       display();
       break;
  }
    printf("Enter choice");
    scanf("%d", &choice);
  }
}
void push(int a){
    if(top == a){
       printf("Stack Overflow condition");
```

```
        return ;
    }
    stack[++pos] = a;
}
int pop() {
    if(pos == -1) {
        printf("Stack Underflow Condition");
        return (int) NULL;
    }
    return stack[pos--];
}
void display() {
    for(int i = 0; i < size; i++) {
        printf("%d", stack[i]);
    }
    printf("\n");
}
```

Output -

1. Push
2. Pop
3. Display
4. Exit

Enter choice : 1
Enter integer to be pushed : 3
Enter choice : 1
Enter integer to be pushed : 4
Enter choice : 2
Integer popped : 4
Enter choice : 1
Enter integer to be pushed : 5
Enter choice : 3
3 5 o o o o o o
Enter choice : 4

24/12/2023
Next week update with Explanation