21/12/23 Lab-2

1) // Program to swap two numbers using pointers

```
#include<stdio.h>

void swap(int *a, int *b){
    int temp = *a;
    *a = *b;
    *b = temp;
}

void main(){
    int a,b;
    printf("Read values for a,b: ");
    scanf("%d%d", &a,&b);
    printf("Value before swapping \n a=%d ; b = %d ");
    swap(&a,&b);
    printf("Value after swapping \n a= %d , b = %d");
}
```

Output:

Read values of a,b: 4 5
Value before swapping
a = 4
b = 5

Value after swapping
a = 5
b = 4

2)
Stack implementation

```
#include <stdio.h>
#include <stdlib.h>

#define max 5

int stack[max];
int top = -1;

void push(int a){
    if (top == max-1){
        printf("Stack is full");
        return;
    }
    top++;
    stack[top] = a;
}

void pop(){
    if (top == -1){
        printf("Stack is empty");
        return;
    }
    top--;
    printf("Lifted element is %d", stack[top+1]);
}
```

```
void display(){
    if (top == -1){
        printf("stack is empty");
        return;
    }
    for(int i=0; i<=top ; i++){
        printf("%d ", stack[i]);
    }

}

void main(){
    int choice,
    int a,b,
    while(1){
        printf("\n select an option: \n1. Push element \n2. Pop Elem
            \n3. Display elements");
        scanf("%d", &choice);
        switch(choices){
            case 1:
                printf("Enter value:");
                scanf("%d",&a);
                push(a);
                break;

            case 2:
                pop();
                break;

            case 3:
                display();
    }
}
```

Output:

Select an option:
1: Push element
2: Pop element
3: Display elements

Choice: 1
Enter value: 2

Choice: 1
Enter value: 4

Choice: 1
Enter value: 3

Choice: 3
2 4 3

Choice: 2
Popped element is 3

: 4

3) // Program to demonstrate dynamic allocation

```
#include<stdio.h>
#include<stdlib.h>

void main(){
    int n, *arr, *arr2;

    printf("Read length of both arrays:");
    scanf("%d", &n);

    arr = (int*)malloc(n*sizeof(int));
    arr2 = (int*) calloc (n, sizeof(int));
    for(int i=0; i<n; i++){
        // arr[i] = i+1;
        arr[i] = n+i+i;
    }

    printf("Array 1 before merging:\n");
    for(int i=0; i<n; i++){
        printf("%d ", arr[i]);
    }

    arr = realloc(arr, 2*n*sizeof(int));

    for(int i=n; i<2*n; i++){
        arr[i] = arr2[i-n];
    }
```

```
for(int i=0; i<2^n; i++){
    printf("%d ", arr[i]);
}
free(arr);
```

```
: 4 5
: 6 7 8 9 10
```