

LAB # 6:

Tasks:

1	<p>Write a program as follows for STACK with 10 integers.</p> <pre> -----Array based implementation of STACK----- 1. Push an element on stack 2. Pop an element from stack 3. Display all 4. Top element 5. Exit ----- Please Enter Your Choice: </pre>
2	<p>Test the program using the following procedure: STACK of size N=6</p> <ol style="list-style-type: none"> 1. Call PUSH(5) 2. Call PUSH(2) 3. Call PUSH(3) 4. Call POP() 5. Call PUSH(6) 6. Call PUSH(9) 7. Call PUSH(3) 8. Call DISPLAY() 9. Call TOP()
3	<p>There are 20 books placed on a table in a pile. Sara wants to arrange these books on shelf. She placed the books on shelf one by one. Write a program that implements the above scenario and print number of books left on table and number of books placed on shelf</p>
4	<p>There are five stacks of Chips packs with different flavours in a carton. Each Stack contains five packs of Chips. Packet weights may vary in each stack. Write a program that implements the above scenario. Print all flavours of chips with each packet weight. Also print the weight of the carton.</p>



5	Ali and Saad are playing a game. Both have list A and B respectively of same numbers (1-20) in different order. Both have to guess the number placed at the last position of the list. If Ali's last number is greater than Saad's last number, then number will be removed from Ali's list and point will be awarded to Ali and same case goes with Saad. Assume that both have 10 turns, and those who have more points will be the winner. Write a program that implements the above scenario
6	Write a program that evaluates the postfix expression using stack.

Task 1 Code:

```
#include<iostream>
using namespace std;
int stack[100], x = 100, top = -1;
void push(int value)
{
    if (top >= x - 1)
        cout << "Stack Overflow \n";
    else
    {
        top++;
        stack[top] = value;
    }
}
void pop()
{
    if (top <= -1)
        cout << "Stack Underflow \n";
    else
    {
        cout << "The Popped Element is " << stack[top] << endl;
        top--;
    }
}
void display()
{
    if (top >= 0)
    {
        cout << "Stack Elements Are: ";
        for (int i = top; i >= 0; i--)
        {
            cout << stack[i] << " ";
        }
        cout << "\n";
    }
    else
    {
        cout << "Stack Is Empty \n ";
    }
}
void topp(){
    cout << " Top element is : ";
    cout << stack[top];
}
```



```
}
void main() {
    int option, position, value;
    do {
        cout << "=====Array Based Implementation of
STACK=====\\n";
        cout << "1. Push an element on stack \\n";
        cout << "2. Pop an element from stack \\n";
        cout << "3. Display all \\n";
        cout << "4. Top Element \\n";
        cout << "0. Exit.\\n";
        cout <<
"-----\\n";
        cout << "Please Enter Your Choice:\\n";
        cin >> option;
        switch (option) {
            case 0:
                break;
            case 1:
                cout << "Enter an Item to Push " << endl;
                cin >> value;
                push(value);
                break;
            case 2:
                pop();
                break;
            case 3:
                cout << "Display Function Called - " << endl;
                display();
                break;
            case 4:
                cout << "Top Element is ";
                topp();
                cout << "\\n";
                break;
        }
    } while (option != 0);
    system("pause");
}
```



Output:

C:\Users\HP\documents\visual studio 2013\Projects\Lab6\Debug\Lab6.exe

```
=====Array Based Implementation of STACK=====
1. Push an element on stack
2. Pop an element from stack
3. Display all
4. Top Element
0. Exit.
```

Please Enter Your Choice:

1

Enter an Item to Push

1

```
=====Array Based Implementation of STACK=====
1. Push an element on stack
2. Pop an element from stack
3. Display all
4. Top Element
0. Exit.
```

Please Enter Your Choice:

1

Enter an Item to Push

2

```
=====Array Based Implementation of STACK=====
1. Push an element on stack
2. Pop an element from stack
3. Display all
4. Top Element
0. Exit.
```

Please Enter Your Choice:

2

The Popped Element is 2

```
=====Array Based Implementation of STACK=====
1. Push an element on stack
2. Pop an element from stack
3. Display all
4. Top Element
0. Exit.
```

```

Please Enter Your Choice:
1
Enter an Item to Push
3
=====Array Based Implementation of STACK=====
1. Push an element on stack
2. Pop an element from stack
3. Display all
4. Top Element
0. Exit.
-----
Please Enter Your Choice:
3
Display Function Called -
Stack Elements Are: 3 1
=====Array Based Implementation of STACK=====
1. Push an element on stack
2. Pop an element from stack
3. Display all
4. Top Element
0. Exit.
-----
Please Enter Your Choice:
4
Top Element is  Top element is : 3
=====Array Based Implementation of STACK=====
1. Push an element on stack
2. Pop an element from stack
3. Display all
4. Top Element
0. Exit.
-----

```

Task 2 Code:

```

#include<iostream>
using namespace std;
int stack[100], n = 6, top = -1;
void push(int value)
{
    if (top >= n - 1)
        cout << "Stack Overflow \n";
    else
    {
        top++;
        stack[top] = value;
    }
}
void pop()
{
    if (top <= -1)
        cout << "Stack Underflow \n";
    else
    {
        cout << "The Popped Element is " << stack[top] << endl;
        top--;
    }
}
void display()
{
    if (top >= 0)

```



```
{
    cout << "Stack Elements Are: ";
    for (int i = top; i >= 0; i--)
    {
        cout << stack[i] << " ";
    }
    cout << "\n";
}
else
{
    cout << "Stack Is Empty \n ";
}
}
void topp(){
    cout << " Top element is : ";
    cout << stack[top];
}
void main() {
    int option, postion, value;
    do {
        cout << "====Array Based Implementation of
STACK====\n";
        cout << "1. Push an element on stack \n";
        cout << "2. Pop an element from stack \n";
        cout << "3. Display all \n";
        cout << "4. Top Element \n";
        cout << "0. Exit.\n";
        cout <<
        "-----\n";
        cout << "Please Enter Your Choice:\n";
        cin >> option;
        switch (option) {
            case 0:
                break;
            case 1:
                cout << "Enter an Item to Push " << endl;
                cin >> value;
                push(value);
                break;
            case 2:
                pop();
                break;
            case 3:
                cout << "Display Function Called - " << endl;
                display();
                break;
            case 4:
                cout << "Top Element is ";
                topp();
                cout << "\n";
                break;
        }
    } while (option != 0);
    system("pause");
}
```

Output:



C:\Users\HP\Documents\Visual Studio 2013\Projects\Lab6\Debug\Lab6.exe

=====**Array Based Implementation of STACK**=====

1. Push an element on stack
2. Pop an element from stack
3. Display all
4. Top Element
0. Exit.

Please Enter Your Choice:

1

Enter an Item to Push

5

=====**Array Based Implementation of STACK**=====

1. Push an element on stack
2. Pop an element from stack
3. Display all
4. Top Element
0. Exit.

Please Enter Your Choice:

1

Enter an Item to Push

2

=====**Array Based Implementation of STACK**=====

1. Push an element on stack
2. Pop an element from stack
3. Display all
4. Top Element
0. Exit.

Please Enter Your Choice:

1

Enter an Item to Push

3

=====**Array Based Implementation of STACK**=====

1. Push an element on stack
 2. Pop an element from stack
 3. Display all
 4. Top Element
 0. Exit.
-



C:\Users\HP\Documents\Visual Studio 2013\Projects\Lab6\Debug\Lab6.exe

```
=====Array Based Implementation of STACK=====
1. Push an element on stack
2. Pop an element from stack
3. Display all
4. Top Element
0. Exit.
-----
Please Enter Your Choice:
3
Display Function Called -
Stack Elements Are: 3 9 6 2 5
=====Array Based Implementation of STACK=====
1. Push an element on stack
2. Pop an element from stack
3. Display all
4. Top Element
0. Exit.
-----
Please Enter Your Choice:
4
Top Element is  Top element is : 3
=====Array Based Implementation of STACK=====
1. Push an element on stack
2. Pop an element from stack
3. Display all
4. Top Element
0. Exit.
-----
Please Enter Your Choice:

```

Task 3 Code:

```
#include<iostream>
using namespace std;
int countshelf;
class stack
{
public:
    int top;
    int countshelf;
    int tarr[20];
    int sarr[20];
public:
    stack()
    {
        top = -1;
        countshelf = 0;
        for (int i = 0; i <19; i++)
        {
            tarr[i] = 20;
        }
        for (int i = 0; i <19; i++)
        {
            sarr[i] = 0;
        }
    }
    void push(int value)
    {
        if (top == 20)
        {
            cout << "Stack Overflow Condition" << endl;
        }
    }
}
```




```

        else
        {
            top++;
            sarr[top] = value;
            countshelf++;
            cout << "Books Shifted : " << " " << countshelf << endl;
            cout << "Books That are Not Shifted : " << 20 - countshelf
<< endl;
        }
    }
};
void main()
{
    stack s;
    cout << "===== Book's Shifting Program =====\n";
    for (int i = 0; i < 14; i++)
    {
        s.push(14);
    }
    system("pause");
}

```

Output:

C:\Users\HP\Documents\Visual Studio 2013\Projects\q3\Debug\q3.exe

```

===== Book's Shifting Program =====
Books Shifted : 1
Books That are Not Shifted : 19
Books Shifted : 2
Books That are Not Shifted : 18
Books Shifted : 3
Books That are Not Shifted : 17
Books Shifted : 4
Books That are Not Shifted : 16
Books Shifted : 5
Books That are Not Shifted : 15
Books Shifted : 6
Books That are Not Shifted : 14
Books Shifted : 7
Books That are Not Shifted : 13
Books Shifted : 8
Books That are Not Shifted : 12
Books Shifted : 9
Books That are Not Shifted : 11
Books Shifted : 10
Books That are Not Shifted : 10
Books Shifted : 11
Books That are Not Shifted : 9
Books Shifted : 12
Books That are Not Shifted : 8
Books Shifted : 13
Books That are Not Shifted : 7
Books Shifted : 14
Books That are Not Shifted : 6
Press any key to continue . . .

```

Task 4 Code:

```

#include<iostream>

using namespace std;

#define Fp 5

```

```
int Carton[Fp], top = -1;
void PUSH(int item){
    if (top == Fp - 1) cout << "Packet fully Filled\n";
    else{ top++; Carton[top] = item; }
}
void POP(){
    if (top <= -1) cout << "Packet is empty\n";
    else top = -1;
}
int weightPacket(){
    int sum = 0;
    if (top >= 0){
        for (int i = 0; i<Fp; i++){
            sum += Carton[i];
        }
    }
    else{
        cout << "Empty Packet\n";
    }
    return sum;
}
void main(){
    int TotalPacket = 6;
    int PacketNo = 0, weight[Fp];
    cout << "===== Weight Calculation Program =====\n";
    do{
        for (int i = 1; i <= 5; i++){
            if (PacketNo == 1){
                PUSH(rand() % 5);
                weight[PacketNo - 1] = weightPacket();
                if (i == 5) POP();
            }
            if (PacketNo == 2){
                PUSH(rand() % 5);
                weight[PacketNo - 1] = weightPacket();
            }
        }
        PacketNo++;
    } while (PacketNo <= TotalPacket);
}
```



```
        if (i == 5) POP();
    }
    if (PacketNo == 3){
        PUSH(rand() % 5);
        weight[PacketNo - 1] = weightPacket();
        if (i == 5) POP();
    }
    if (PacketNo == 4){
        PUSH(rand() % 5);
        weight[PacketNo - 1] = weightPacket();
        if (i == 5) POP();
    }
    if (PacketNo == 5){
        PUSH(rand() % 5);
        weight[PacketNo - 1] = weightPacket();
        if (i == 5) POP();
    }
}
PacketNo++;
} while (PacketNo != 6);
int TotalCartonWeight = 0;
for (int i = 0; i < Fp; i++){
    cout << "Weight Of Packet No " << i + 1 << " : " << weight[i] << "
kg\n";
    TotalCartonWeight += weight[i];
}
cout << "Total weight of flavor Carton : " << TotalCartonWeight << " kg"
<< endl;
system("pause");
}
```

Output:

```
C:\Users\HP\Documents\Visual Studio 2013\Projects\Lab6\Debug\Lab6.exe

===== Weight Calculation Program =====
Weight Of Packet No 1 : 11 kg
Weight Of Packet No 2 : 16 kg
Weight Of Packet No 3 : 4 kg
Weight Of Packet No 4 : 6 kg
Weight Of Packet No 5 : 12 kg
Total weight of flavor Carton : 49 kg
Press any key to continue . . .
```

Task 5 Code:

```
#include<iostream>
using namespace std;
int stack1[10] = { 1, 19, 17, 15, 3, 9, 11, 5, 7, 13 };
int stack2[10] = { 10, 2, 4, 6, 8, 12, 14, 16, 18, 20 };
int x = 10, top = -1, top2 = -1;
void pop1()
{
    if (top <= -1)
        cout << "Stack Underflow \n";
    else
    {
        cout << "The Popped Element is " << stack1[top] << endl;
        top--;
    }
}
void pop2()
{
    if (top2 <= -1)
        cout << "Stack Underflow \n";
    else
    {
        cout << "The Popped Element is " << stack2[top2] << endl;
        top2--;
    }
}
void main()
{
    cout << "Numbers of Ali is ";
    for (int i = 0; i < 10; i++)
    {
        cout << stack1[i] << " , ";
    }
    cout << endl;
    cout << "Numbers of Saad is ";
    for (int i = 0; i < 10; i++)
    {
        cout << stack2[i] << " , ";
    }
    cout << endl;
    int count1 = 0, count2 = 0;
    for (int i = 0; i < 10; i++)
    {
        if (stack1[i] > stack2[i])
        {
            count1++;
        }
    }
}
```



```
        pop1();
    }
    if (stack1[i] < stack2[i])
    {
        count2++;
        pop2();
    }
}
if (count1 > count2)
{
    cout << "Ali is Winner with " << count1 << " points\n";
}
if (count1 < count2)
{
    cout << "Saad is Winner with " << count2 << " points\n";
}
system("pause");
}
```

Output:

```
C:\Users\HP\Documents\Visual Studio 2013\Projects\Lab6\Debug\Lab6.exe
Numbers of Ali is 1 , 19 , 17 , 15 , 3 , 9 , 11 , 5 , 7 , 13 ,
Numbers of Saad is 10 , 2 , 4 , 6 , 8 , 12 , 14 , 16 , 18 , 20 ,
Stack Underflow
Stack Underflow
Stack Underflow
Stack Underflow
Stack Underflow
Stack Underflow
Stack Underflow
Stack Underflow
Stack Underflow
Stack Underflow
Stack Underflow
Saad is Winner with 7 points
Press any key to continue . . .
```

Task 6 Code:

```
#include <iostream>
#include <string>
using namespace std;
struct Stack
{
    int top;
    unsigned capacity;
    int* array;
};

struct Stack* createStack(unsigned capacity)
```

```
{
    struct Stack* stack = (struct Stack*) malloc(sizeof(struct Stack));
    if (!stack) return NULL;
    stack->top = -1;
    stack->capacity = capacity;
    stack->array = (int*)malloc(stack->capacity * sizeof(int));
    if (!stack->array) return NULL;
    return stack;
}

int isEmpty(struct Stack* stack)
{
    return stack->top == -1;
}

char peek(struct Stack* stack)
{
    return stack->array[stack->top];
}

char pop(struct Stack* stack)
{
    if (!isEmpty(stack))
        return stack->array[stack->top--];
    return '$';
}

void push(struct Stack* stack, char op)
{
    stack->array[++stack->top] = op;
}

int evaluatePostfix(char* exp)
{
    struct Stack* stack = createStack(strlen(exp));
    int i;
    if (!stack) return -1;
    for (i = 0; exp[i]; ++i)
```



```
{
    if (isdigit(exp[i]))
        push(stack, exp[i] - '0');
    else
    {
        int val1 = pop(stack);
        int val2 = pop(stack);
        switch (exp[i])
        {
            case '+': push(stack, val2 + val1); break;
            case '-': push(stack, val2 - val1); break;
            case '*': push(stack, val2 * val1); break;
            case '/': push(stack, val2 / val1); break;
        }
    }
}

return pop(stack);
}

void main()
{
    cout << "=====Postfix Evaluation Program=====\\n";
    char exp[] = "455*+ 8-";
    cout << "\\nPostfix Evaluation " << "of Expression " << exp << " is = "
    << evaluatePostfix(exp);
    cout << "\\n\\n";
    system("pause");
}
```

Output:



DEPARTMENT OF
COMPUTER
SCIENCES

CSC-221: Data Structures and Algorithms

Mohammad Saad Bin Younus

02-134201-041

BSCS - 3(A)

C:\Users\HP\Documents\Visual Studio 2013\Projects\Lab6\Debug\Lab6.exe

=====Postfix Evaluation Program=====

Postfix Evaluation of Expression 455*+ 8- is = 28

Press any key to continue . . . _

