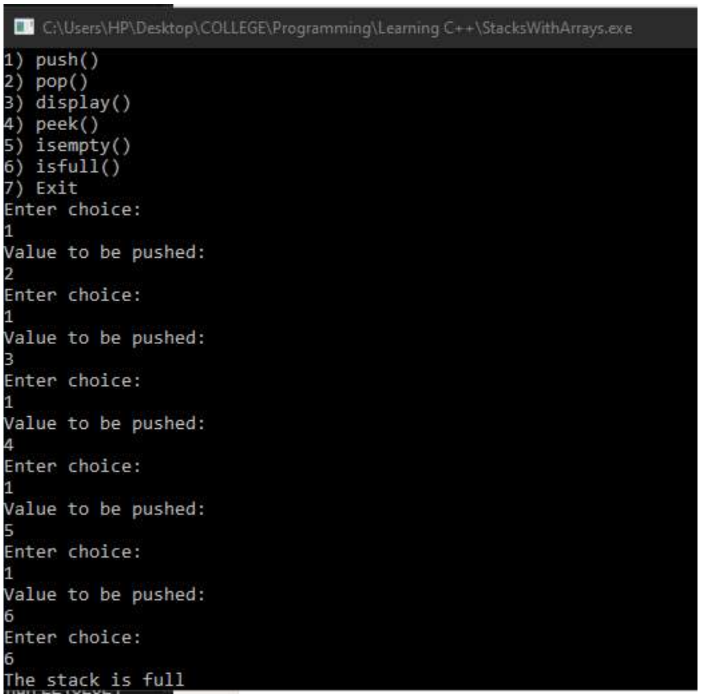
Week1 Batch3

1. Implementation of Stack using arrays

```
#include<iostream>
using namespace std;

int stack[5],n=5,top=-1;

void push(int value){

if(top>=n-1)
cout<<"Stack Overflow"<<endl;

else{
top++;
stack[top]=value;
}
}

void pop(){

if(top<=-1)
cout<<"Stack Underflow"<<endl;

else{
cout<<"Popped Element:"<< stack[top]<<endl;
top--;
}

}

void display(){

if(top>=0) {
cout<<"Stack elements:";
for(int i=top; i>=0; i--) {
cout<<stack[i]<<" ";
cout<<endl;
}
}

else
cout<<"Stack is empty"<<endl;

}
void peek(){
   if(top>-1)
   {
      cout<<"Top element: "<<stack[top]<<endl;
   }
   else
      cout<<"Empty stack"<<endl;
}
```

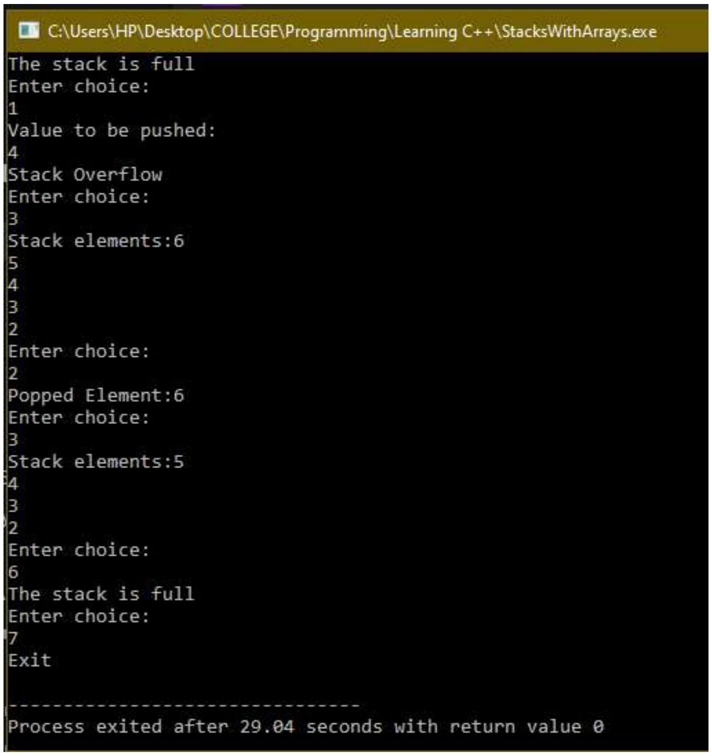
```
void isEmpty(){
  if(top==-1)
    cout<<"The stack is empty"<<endl;
  else
    cout<<"The stack is not empty"<<endl;
}



void isFull(){
  if(top==n)
    cout<<"The stack is not full"<<endl;
  else
    cout<<"The stack is full"<<endl;
}



int main(){
int c, value;
cout<<"1) push()"<<endl;
cout<<"2) pop()"<<endl;
cout<<"3) display() "<<endl;
cout<<"4) peek()"<<endl;
cout<<"5) isempty()"<<endl;
cout<<"6) isfull()"<<endl;
cout<<"7) Exit"<<endl;

do {

cout<<"Enter choice: "<<endl;
cin>>c;
switch(c) {

case 1: {
cout<<"Value to be pushed:"<<endl;
cin>>value;
push(value);
break;
}

case 2: {
pop();
break;
}

case 3: {
display();
break;
}

case 4:{
peek();
break;
}

case 5:{
```

```
isEmpty();
break;
}

case 6 :{
isFull();
break;
}

case 7: {
cout<<"Exit"<<endl;
break;
}

default: {cout<<"Invalid Choice"<<endl;

}

}

}while(c!=7);return 0;

}
```



C:\Users\HP\Desktop\COLLEGE\Programming\Learning C++\StacksWithArrays.exe

```
1) push()
2) pop()
3) display()
4) peek()
5) isempty()
6) isfull()
7) Exit
Enter choice:
1
Value to be pushed:
2
Enter choice:
1
Value to be pushed:
3
Enter choice:
1
Value to be pushed:
4
Enter choice:
1
Value to be pushed:
5
Enter choice:
1
Value to be pushed:
6
Enter choice:
6
The stack is full
```

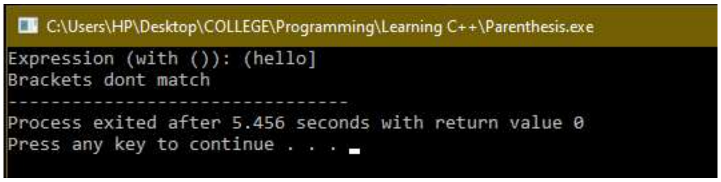


C:\Users\HP\Desktop\COLLEGE\Programming\Learning C++\StacksWithArrays.exe

```
The stack is full
Enter choice:
1
Value to be pushed:
4
Stack Overflow
Enter choice:
3
Stack elements:6
5
4
3
2
Enter choice:
2
Popped Element:6
Enter choice:
3
Stack elements:5
4
3
2
Enter choice:
6
The stack is full
Enter choice:
7
Exit

--------------------------------
Process exited after 29.04 seconds with return value 0
```

2. To check if the given parenthesized expression has properly matching open and closing parenthesis

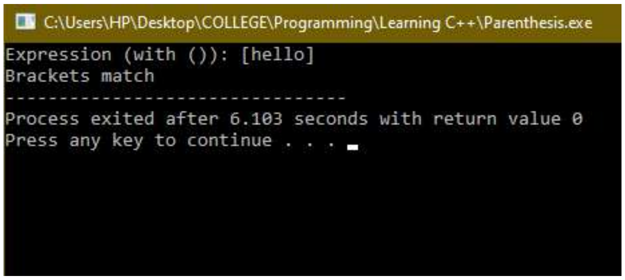
```
#include<iostream>
using namespace std;

int stack[100], n=100, top= -1;
```



C:\Users\HP\Desktop\COLLEGE\Programming\Learning C++\Parenthesis.exe

```
Expression (with ()): (hello]
Brackets dont match
--------------------------------
Process exited after 5.456 seconds with return value 0
Press any key to continue . . .
```

```
void push(int value){

if(top>=n-1)
cout<<"Stack Overflow"<<endl;

else{
top++;
stack[top]=value;
}
}

void pop(){

if(top<=-1) {
cout<<"Stack Underflow"<<endl;
cout<<"Brackets dont match"<<endl;
exit(0);

}

else{
top--;
}

}

bool isEmpty()
{
if (top==-1){
return true;
}

else {
return false;
}

}

bool matching (string s)
{
char t;
for (int i=0; i<s.length();i++){

if ( s[i]=='(')
{
push(s[i]);
continue;
}

if (s[i]==')')
{
pop();
}

}
return (isEmpty());

}
```

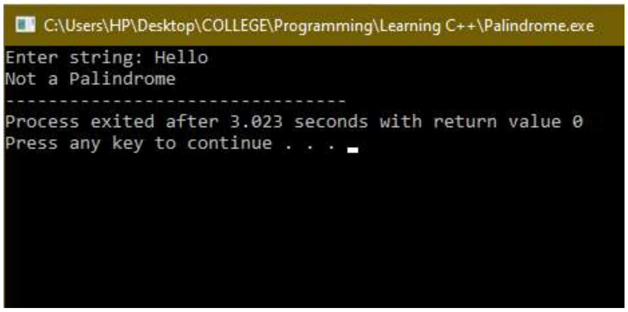

C:\Users\HP\Desktop\COLLEGE\Programming\Learning C++\Parenthesis.exe

```
Expression (with ()): [hello]
Brackets match
--------------------------------
Process exited after 6.103 seconds with return value 0
Press any key to continue . . .
```

```
int main()
{
  string exp;
  cout<<"Expression (with ()): ";
  cin>>exp;
  if (matching(exp))
    cout << "Brackets match";
  else
    cout << "Brackets dont match";
  return 0;
}
```

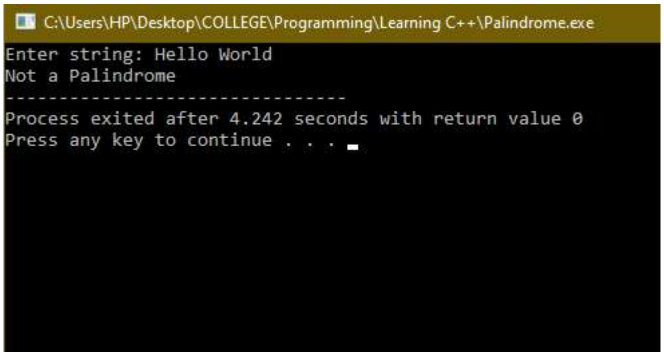
3. To check a given string is palindrome or not using stack

```
#include <iostream>
#include <string>

using namespace std;

int n=50, top=-1;
char stack[50];

void push(char x)
{
  top++;
  stack[top]=x;
}



void pop()
{
  if(top==-1)
    cout<<"Stack Underflow"<<endl;
  else
  {
    top--;
  }
}
char peek()
{
  return stack[top];
}



bool Pal(string s)
{
  int length, k=1;
  while(s[k]!='\0')
  {
  k++;
}
length=k;
```



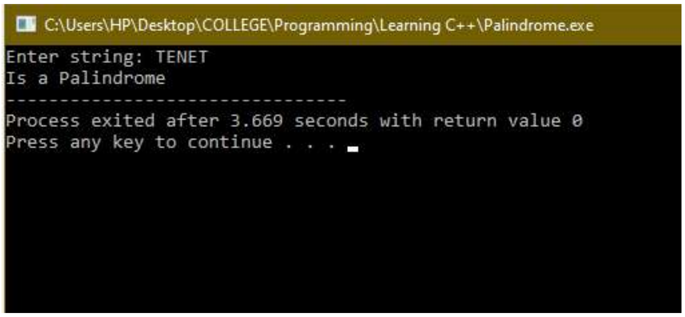
C:\Users\HP\Desktop\COLLEGE\Programming\Learning C++\Palindrome.exe

```
Enter string: Hello
Not a Palindrome
--------------------------------
Process exited after 3.023 seconds with return value 0
Press any key to continue . . .
```



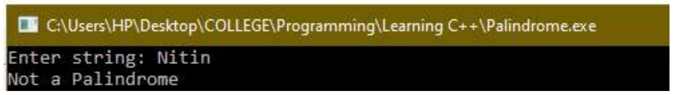
C:\Users\HP\Desktop\COLLEGE\Programming\Learning C++\Palindrome.exe

```
Enter string: Hello World
Not a Palindrome
--------------------------------
Process exited after 4.242 seconds with return value 0
Press any key to continue . . .
```



C:\Users\HP\Desktop\COLLEGE\Programming\Learning C++\Palindrome.exe

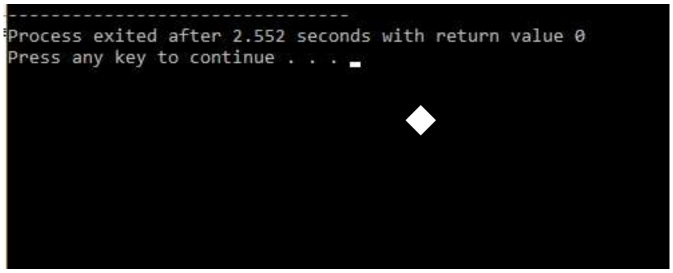
```
Enter string: TENET
Is a Palindrome
--------------------------------
Process exited after 3.669 seconds with return value 0
Press any key to continue . . .
```



C:\Users\HP\Desktop\COLLEGE\Programming\Learning C++\Palindrome.exe

```
Enter string: Nitin
Not a Palindrome
```

```
  int i, mid = length / 2;
  for (i = 0; i < mid; i++) {
    push(s[i]);
  }
  if (length % 2 != 0) {
    i++;
  }
  char m;
  while (s[i] != '\0')
  {
    m = peek();
    pop();
    if (m != s[i])
      return false;
    i++;
  }

  return true;
}
```

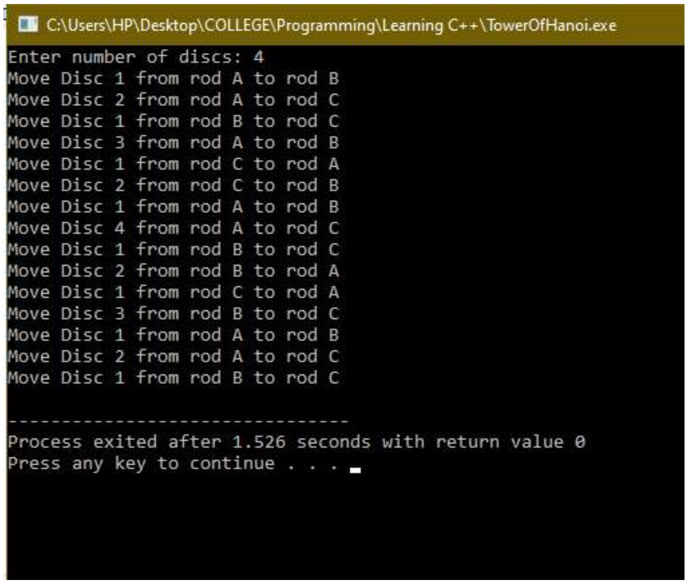


```
--------------------------------
Process exited after 2.552 seconds with return value 0
Press any key to continue . . .
```

```
int main()
{
  string s;
  cout<<"Enter string: ";
  cin>>s;
  if (Pal(s)) {
    cout << "Is a Palindrome";
  }
  else {
    cout << "Not a Palindrome";
  }
  return 0;
}
```

4. Tower of Hanoi for n disks(Recursion application)

```
#include<iostream>
using namespace std;

void tOfHanoi ( int i, char f_r, char t_r, char aux_r )
{
if (i==1){
cout<<"Move Disc 1"<<" from rod "<<f_r<<" to rod "<<t_r<<endl;
return;

}
tOfHanoi( i-1, f_r, aux_r, t_r);
  cout<<"Move Disc "<<i<<" from rod "<<f_r<<" to rod "<<t_r<<endl;
  tOfHanoi(i-1, aux_r, t_r, f_r);
}
```



C:\Users\HP\Desktop\COLLEGE\Programming\Learning C++\TowerOfHanoi.exe

```
Enter number of discs: 4
Move Disc 1 from rod A to rod B
Move Disc 2 from rod A to rod C
Move Disc 1 from rod B to rod C
Move Disc 3 from rod A to rod B
Move Disc 1 from rod C to rod A
Move Disc 2 from rod C to rod B
Move Disc 1 from rod A to rod B
Move Disc 4 from rod A to rod C
Move Disc 1 from rod B to rod C
Move Disc 2 from rod B to rod A
Move Disc 1 from rod C to rod A
Move Disc 3 from rod B to rod C
Move Disc 1 from rod A to rod B
Move Disc 2 from rod A to rod C
Move Disc 1 from rod B to rod C

--------------------------------
Process exited after 1.526 seconds with return value 0
Press any key to continue . . .
```

```
int main(){
    int i;
    cout<<"Enter number of discs: ";
    cin>>i;
    tOfHanoi(i, 'A', 'C', 'B');
    return 0;
}
```