# Experiment Number 15

**Aim:** Implement any one storage allocation strategies (heap, stack, static).

**Algorithm:**

Step 1: Start

Step 2: Read choice.

Step 3: Perform desired operation.

Step 4: Print respective output.

Step 5: Repeat until user is done.

Step 6: Stop.

**Code:**

```
#include <iostream>
using namespace std;
int main()
{
   string stack[10];
   int top=-1;
   char cont='y';
   while(cont=='y'){
      int c;
      cout<<"Enter :\n1 - To Display contents of stack\n2 - To push an element into
the stack\n3 - To pop an element from the stack\n4 - Display top element of stack\n";
      cout<<"Enter your choice (1-4): ";
      cin>>c;
      switch(c){
         case 1:
         if(top==-1)
         cout<<"STACK IS EMPTY!!\n";
         else
         for(int i=top;i>=0;i--)
         cout<<stack[i]<<endl;
         break;
         case 2:
         if(top==9)
         cout<<"STACK OVERFLOW!!\n";
         else{
         cout<<"Enter the element: ";
         cin>>stack[++top];}
         break;
         case 3:
```

```
        if(top==-1)
        cout<<"STACK UNDERFLOW!!\n";
        else
        cout<<"Popped Element: "<<stack[top--]<<endl;
        break;
        case 4:
        cout<<"Top Element: "<<stack[top]<<endl;
        break;
        default:
        cout<<"Invalid Choice!!";
      }
      cout<<"Do you want to continue?(y/n): ";
      cin>>cont;
   }cout<<"\t***";
   return 0;
}
```

**Output:**

```
Enter :
1 - To Display contents of stack
2 - To push an element into the stack
3 - To pop an element from the stack
4 - Display top element of stack
Enter your choice (1-4): 2
Enter the element: a=B=c;
Do you want to continue?(y/n): y
Enter :
1 - To Display contents of stack
2 - To push an element into the stack
3 - To pop an element from the stack
4 - Display top element of stack
Enter your choice (1-4): 3
Popped Element: a=B=c;
Do you want to continue?(y/n): y
Enter :
1 - To Display contents of stack
2 - To push an element into the stack
3 - To pop an element from the stack
4 - Display top element of stack
Enter your choice (1-4): 1
STACK IS EMPTY!!
Do you want to continue?(y/n): n
        ***
```

**Result:** Thus, Stack implemented successfully.