```
//Reallocation of memory
#include<iostream>
#include<stdlib.h>
using namespace std;
int main()
{
int n,m;
cout<<"\n Enter the initial size:";
cin>>n;
int *arr=new int[n];
if(arr==NULL)
{
cout<<"\nMemory allocation failed";
exit(1);
}
cout<<"\nEnter the array elements:";
for(int i=0;i<n;i++)
{
cin>>arr[i];
}
cout<<"\nThe array elements are:";
for(int i=0;i<n;i++)
{
cout<<" "<<arr[i];
}
cout<<"\n\n Memory requirement increases:";
cout<<"\nEnter new size(greater than n))";
cin>>m;
int *temp=new int[m];
if(temp==NULL)
{
cout<<"\n\n\nMemory Allocation failed";
exit(1);
}
cout<<"\n\n\n COPYING THE OLD ARRAY";
for(int i=0;i<n;i++)
{
temp[i]=arr[i];
}
delete []arr;
arr=temp;
cout<<"\n Enter new values:";
for(int i=n;i<m;i++)
{
cin>>arr[i];
}
cout<<"\n The array elements are:";
for(int i=0;i<m;i++)
{
```

```
cout<<" "<<arr[i];
}
delete [] arr;
return 0;
}
---------------------------------------
#include<iostream>
#include<stdlib.h>
using namespace std;
int main()
{
int n,m,i,key,flag=-1;
cout<<"\n Enter the initial size:";
cin>>n;
int *arr=new int[n];
if(arr==NULL)
{
cout<<"\nMemory allocation failed";
exit(1);
}
cout<<"\nEnter the array elements:";
for(i=0;i<n;i++)
{
cin>>arr[i];
}
cout<<"\nEnter element to search:";
cin>>key;
for(i=0;i<n;i++)
{
  if(arr[i]==key)
  {
     flag=i;
     break;
  }
}
if(flag!=-1)
{
   cout<<"\nElement found at index:"<<flag;
}
else
{
cout<<"\n\n Memory requirement increases:";
cout<<"\nEnter new size(greater than n))";
cin>>m;
int *temp=new int[m];
if(temp==NULL)
{
cout<<"\n\n\nMemory Allocation failed";
exit(1);
```

```
}
cout<<"\n\n\n COPYING THE OLD ARRAY";
for(i=0;i<n;i++)
{
temp[i]=arr[i];
}
delete []arr;
arr=temp;
cout<<"\n Enter new values:";
for(i=n;i<m;i++)
{
cin>>arr[i];
}
for(i=n;i<m;i++)
{
  if(arr[i]==key)
  {
     flag=i;
     break;
  }
}
if(flag==-1)
{
   cout<<"\nElement not found";
}
else
{
   cout<<"\nElement found at index:"<<flag;
}
}
delete [] arr;
return 0;
}
```

---

```
#include<iostream>
using namespace std;
class Array
{
private:
int *arr;
int n;
public:
Array();
void show_data();
};
Array::Array()//Dynamic constructor
{
cout<<"\nEnter size:";
cin>>n;
```

```
arr=new int[n];
cout<<"\n Enter the elements:";
for(int i=0;i<n;i++)
{
cin>>arr[i];
}
}
void Array::show_data()
{
for(int i=0;i<n;i++)
{
cout<<" "<<arr[i];
}
}
int main()
{
int no_object;
cout<<"\n Enter no. of objects:";
cin>>no_object;
Array *ptr=new Array[no_object];
Array *ptr1=ptr;
for(int i=0;i<no_object;i++)
{
ptr->show_data();
ptr++;
}
delete [] ptr1;
cout<<"\nMemory deallocated successfully";
return 0;
}
---------------------------------------
#include<iostream>
using namespace std;
class Array
{
private:
double *arr;
int n;
public:
Array();
void task();
};
Array::Array()//Dynamic constructor
{
cout<<"\nEnter size:";
cin>>n;
arr=new double[n];
cout<<"\n Enter the elements:";
for(int i=0;i<n;i++)
```

```
{
cin>>arr[i];
}
}
void Array::task()
{
double sum=0.0,avg=0.0;
for(int i=0;i<n;i++)
{
sum=sum+arr[i];
}
avg=sum/n;
cout<<"\nAverage is:"<<avg;
}
int main()
{
int no_object;
cout<<"\n Enter no. of objects:";
cin>>no_object;
Array *ptr=new Array[no_object];
Array *ptr1=ptr;
for(int i=0;i<no_object;i++)
{
ptr->task();
ptr++;
}
delete [] ptr1;
cout<<"\nMemory deallocated successfully";
return 0;
}
```

---

```
// Virtual destructor

#include<iostream>
using namespace std;
class base {
  public:
    base()
    { cout<<"Constructing base \n"; }
    virtual~base()
    { cout<<"Destructing base \n"; }
};
class derived1: public base {
  public:
    derived1()
    { cout<<"Constructing derived \n"; }
    ~derived1()
    { cout<<"Destructing derived \n"; }
};
```

```
int main()
{
  base *b = new derived1;
  delete b;
  return 0;
}
-------------------------------------
#include <iostream>
#include <string.h>       //for strcpy(), etc
using namespace std;
class string1             //user-defined string type
   {
   private:
      char* str;                  //pointer to string
   public:
      string1(char* s)              //constructor, one arg
         {
         int length = strlen(s);    //length of string argument
         str = new char[length+1];  //get memory
         strcpy(str, s);            //copy argument to it
         }
      ~string1()                    //destructor
         {
         cout << "Deleting str\n";
         delete[] str;          //release memory
         }
         void display()          //display the String
         {
         cout << str << endl;
         }
   };
int main()
   {                    //uses 1-arg constructor
   string1 s1("string example");
   cout << "s1=";                  //display string
   s1.display();
   return 0;
   }
   ------------------------------
   #include<iostream>
using namespace std;
class s
{
        int rollno;
        char name[10];
        int age;
        s *next;
        public:
```

```
        s()
        {
            next =NULL;
        }
        void getdata()
        {
            cout<<"Enter rollno, name, age  "<<endl;
            cin>>rollno>>name>>age;
        }
        void link(s *t)
        {
            next=t;
        }
        void print()
        {
            s *p=this;
            while(p!=NULL)
            {
cout<<"Roll no  = "<<p->rollno<<" Name =  "<<p->name<<" Age  ="<<p->age<<endl;
                p=p->next;
            }
        }
};
int main()
{
    s o1,o2,o3;
    o1.getdata();
    o2.getdata();
    o3.getdata();
    o1.link(&o2);
    o2.link(&o3);
    o1.print();
    return 0;
}
```