Practical-4.(b)

Aim: Write a C++ program to overload new/delete operators in a class.

Algorithm: (i) Start

(ii)void*operator new(size_t size)

(iii)void operator delete(void*)

(iv)Stop

Theory: The new and delete operators can also be overloaded like other operators in C++. New and Delete operators can be overloaded globally or they can be overloaded for specific classes.

Input:

```
#include<iostream>
using namespace std;
void * operator new(size_t size)
{
    cout << "New operator overloading " << endl;
    void * p = malloc(size);
    return p;
}
```

```
void operator delete(void * p)
{
    cout << "Delete operator overloading " << endl;
    free(p);
}
int main()
{
    std::cout<<"08_Rabin Nadar"<<std::endl;
    int n = 5, i;
    int * p = new int[n];

    for (i = 0; i<n; i++)
    p[i]= i;

    cout << "Array: ";
    for(i = 0; i<n; i++)
        cout << p[i] << " ";

    cout << endl;
```

```
    delete [] p;
}
```

Output:

Output Clear

```
/tmp/ZMJHVaDakl.o
08_Rabin Nadar
New operator overloading
Array: 0 1 2 3 4
Delete operator overloading
```