

**Week 1 (Aug 31st – Sept 4 th )**  
**Topics: Dynamic Memory allocation in C++**

1)D) All 1, 2 & 3

2)

Array p: 5 7 11 17 25

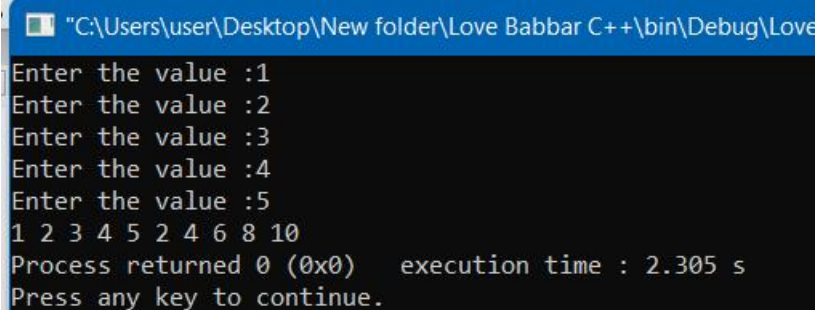
Array q: 25 17 11 7 5

3) Write a C++ program to perform Insertion and deletion operation in array by creating array of marks of students in a class using dynamic memory allocation (new and delete operator).

```
#include <iostream>
using namespace std;
int main()
{
    int *a;
    a = new int [5]; A
    for(int i=0;i<5;i++)
    {
        cin>>a[i];
    }
    delete []a;
    return 0;
}
```

4)Write a C++ program to create a new array that is twice the size of the argument array. The functions should copy the contents of the argument array to the first half of the new array, and the contents of the argument array each multiplied by 2 to the second half of the new array. The function should return a pointer to the new array

```
#include <iostream>
using namespace std;
int main()
{
    int *a;
    a = new int [5];
    for(int i=0;i<5;i++)
        cin>>a[i];
    int *b,n=10;
    b = new int [n];
    for(int i=0;i<n/2;i++)
    {
        b[i]=a[i];
    }
    for(int i=n/2;i<n;i++)
    {
        b[i]=2*a[i-n/2];
    }
    for(int i=0;i<n;i++)
        cout<<b[i]<<" ";
    delete []a;
    delete []b;
    return 0;
}
```



```
"C:\Users\user\Desktop\New folder\Love Babbar C++\bin\Debug\Love
Enter the value :1
Enter the value :2
Enter the value :3
Enter the value :4
Enter the value :5
1 2 3 4 5 2 4 6 8 10
Process returned 0 (0x0) execution time : 2.305 s
Press any key to continue.
```

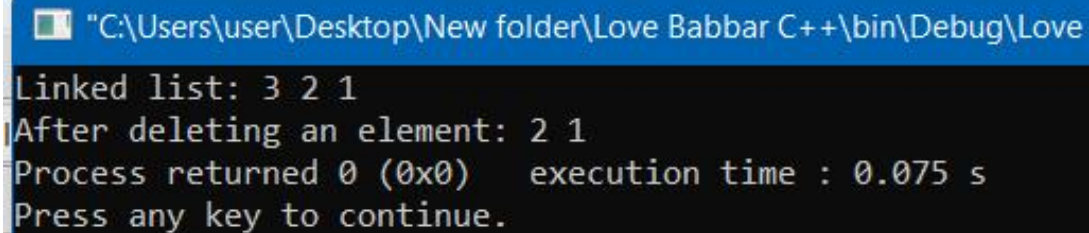
5) WAP in C++ using new and delete operator. The structure Node forms the linked list node. It contains the data and a pointer to the next linked list node. The function insertNode() inserts the data into the beginning of the linked list. It creates a new\_node and inserts the number in the data field of the newNode. Then the new\_node points to the head. Finally the head is the newNode i.e. the linked list starts from there. The function display() displays the whole linked list. First temp points to head. Then it is continuously forwarded to the next node until all the data values of the nodes are printed. deleteItem() function deletes the items from the list. In the function main(), first various values are inserted and deleted into the linked list by calling insertNode() and deleteItem(). Then the linked list is displayed.

```
#include <iostream>
using namespace std;
struct Node {
int data;
struct Node* next;
}*head=NULL;
void insertNode( int new_data) {
struct Node* new_node = new Node();
new_node->data = new_data;
new_node->next = head;
head = new_node;
}
void deleteItem( int key) {
struct Node *temp = head, *prev;
if (temp != NULL && temp->data == key) {
head = temp->next;
free(temp);
return;
}
while (temp != NULL && temp->data != key) {
prev = temp;
temp = temp->next;
}
if (temp == NULL) return;
```

```

prev->next = temp->next;
free(temp);
}
void display(struct Node*p) {
while (p!= NULL) {
cout <<p->data << " ";
p =p->next;
}
}
int main() {
insertNode(1);
insertNode(2);
insertNode(3);
cout << "Linked list: ";
display(head);
cout << "\nAfter deleting an element: ";
deleteltem( 3);
display(head);
}

```



```

"C:\Users\user\Desktop\New folder\Love Babbar C++\bin\Debug\Love
Linked list: 3 2 1
After deleting an element: 2 1
Process returned 0 (0x0)   execution time : 0.075 s
Press any key to continue.

```