***NAME : Himanshu Dixit***

***ENROLL NO. : 21103262***

***BATCH : B11***

***Data Structure [15B11CI311]***

***Tutorial Sheet***

***Week 1***

***Q1)*** *Which of the following is true about new when compared with malloc.*

*1) new is an operator, malloc is a function*

*2) new calls constructor, malloc doesn't*

*3) new returns appropriate pointer, malloc returns void \* and pointer needs to typecast to  appropriate type.*

*a) 1 &2*

*b) 1 & 3*

*c) 2 & 3*

***d)*** *All 1, 2 & 3*

***Solution :***

***d)*** *All 1,2 & 3*

***Q2)*** *What is the output of the following code?*

*int \*p;*

*int \*q;*

*p = new int [5];*

*p[0] = 5;*

*for (int i = 1; i < 5; i++)*

*p[i] = p[i - 1] + 2 \* i;*

*cout << "Array p: ";*

*for (int i = 0; i < 5; i++)*

*cout << p[i] << " ";*

*cout << endl;*

*q = new int[5];*

*for (int i = 0; i < 5; i++)*

*q[i] = p[4 - i];*

*cout << "Array q: ";*

*for (int i = 0; i < 5; i++)*

*cout << q[i] << " ";*

*cout << endl;*

***Solution :***

*5 7 11 17 25*

*25 17 11 7 5*

***Q3)*** *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).*

***Solution :***

#include <bits/stdc++.h>

using namespace std;

int main()

{

int n;

cin >> n;

int \*arr = new int[n+1];

for (int i = 0; i < n; i++)

{

cin >> arr[i];

}

int num, pos;

cout << "Enter Number and position from where you wants to store : ";

cin >> num >> pos;

for (int i = n - 1; i >= pos - 1; i--)

{

arr[i + 1] = arr[i];

}

arr[pos - 1] = num;

for (int i = 0; i < n + 1; i++)

{

cout << arr[i] << " ";

}

cout << endl;

int del;

cout << "Enter position you wants to delete from array : ";

cin >> del;

for (int i = del; i < n+1; i++)

{

arr[i - 1] = arr[i];

}

for (int i = 0; i < n ; i++)

{

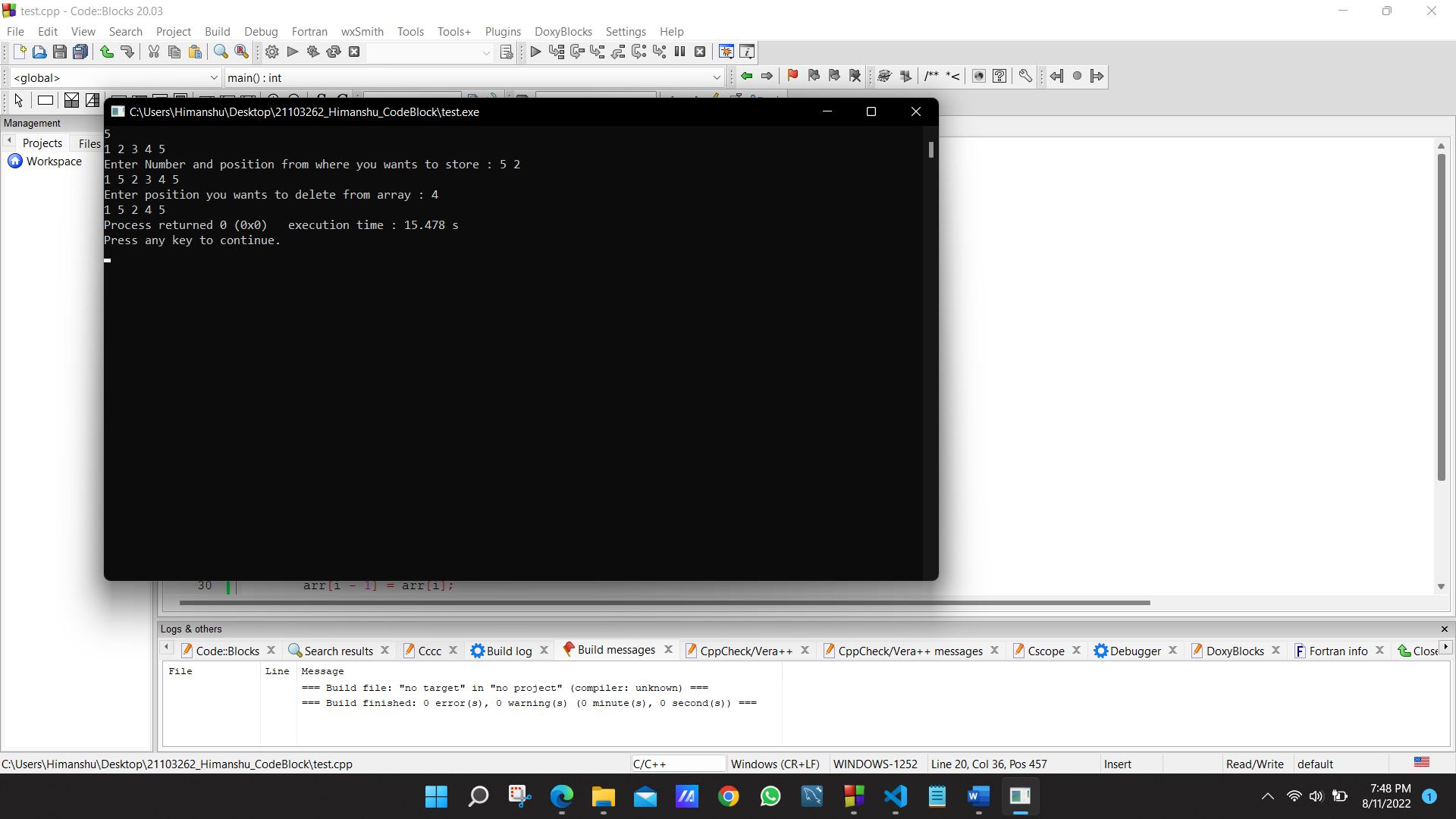
cout << arr[i] << " ";

}

delete []arr;

return 0;

}



***Q4)*** *Write a C++ program to create a new array that is twice the size of the argument array. The  function 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.*

***Solution :***

#include <bits/stdc++.h>

using namespace std;

int main()

{

int n;

cin >> n;

int a[n];

for (int i = 0; i < n; i++)

{

cin >> a[i];

}

int b[2 \* n], i = 0;

for (i = 0; i < n; i++)

{

b[i] = a[i];

}

for (i = n; i < 2 \* n; i++)

{

b[i] = 2\*a[i-n];

}

for (i = 0; i < 2 \* n; i++)

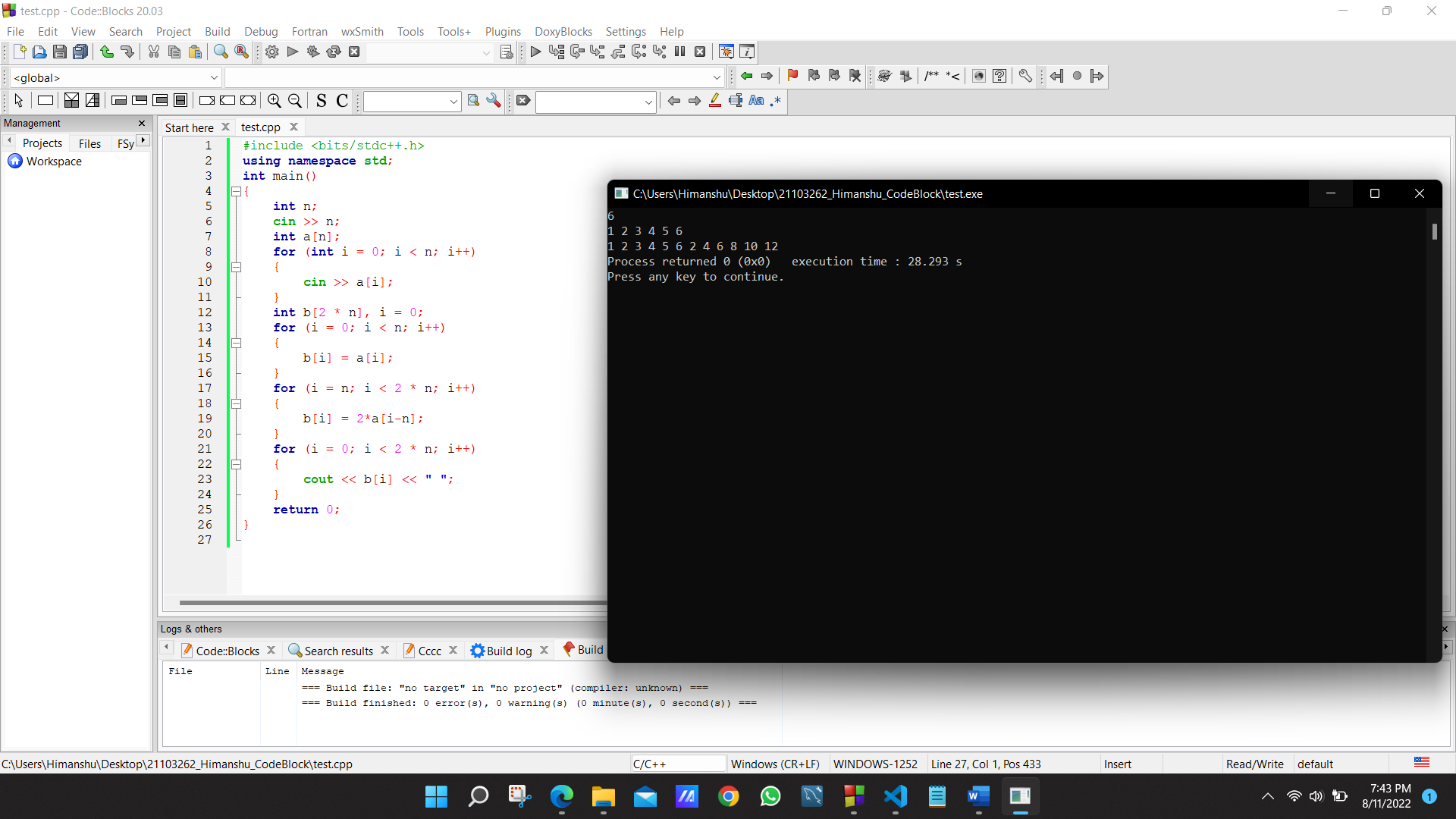
{

cout << b[i] << " ";

}

return 0;

}



***Q5)*** *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.*

***Solution :***

#include <iostream>

using namespace std;

class Node

{

int data;

Node \*next;

public:

Node(int a)

{

data = a;

next = NULL;

}

Node \*insertnode(int a)

{

Node \*temp = new Node(a);

temp->next = this;

return temp;

}

void display()

{

Node \*temp = this;

while (temp != NULL)

{

cout << temp->data << " ";

temp = temp->next;

}

}

Node \*deleteItem(int a)

{

if (a == 0)

{

return this->next;

}

Node \*temp = this;

while (--a)

{

temp = temp->next;

}

if (temp->next->next == NULL)

{

temp->next = NULL;

}

else

{

temp->next = temp->next->next;

}

return this;

}

};

int main()

{

Node \*a = new Node(8);

a = a->insertnode(7);

a = a->insertnode(6);

a = a->insertnode(5);

a = a->insertnode(4);

a = a->insertnode(3);

a = a->insertnode(2);

a = a->insertnode(1);

a = a->insertnode(0);

a = a->deleteItem(0);

a->display();

return 0;

}

