**GDB**

In the working directory, you will find a program that has some error associated with it. The file contains a description of what the code in the file expects to achieve.

Use GDB to figure out the error/errors in each file and correct them.

The working directory contains a folder for test cases. Run each file against the corresponding input, and the generated output should match the expected output after resolving all bugs.

**Question:**

/\*

This program takes as input the number of elements in an array. It then takes an input of all the elements of the array and sorts the given array in ascending order using bubble sort and prints it.

\*/

#include<iostream>

typedef int64\_t ll;

using namespace std;

int main()

{

ll i, n, j, temp;

cin>>n;

ll arr[n];

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

cin>>arr[i];

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

{

for(j=0; j<i; j++) // This loop should run till j<n-1

{

if(arr[j]<arr[j+1]) // This comparison should be greater than for sorting in ascending order

{

temp = arr[j];

arr[j+1] = temp; // For swapping arr[j] should have the value of arr[j+1] first

arr[j] = arr[j+1];

}

}

}

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

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

return 0;

}

Example test cases:

Test case 1:

5

4 2 6 1 9

Expected output:

1 2 4 6 9

Test case 2:

4

6 -5 2 3

Expected output:

-5 2 3 6

**Question 1:**

/\*

This program takes as input the number of rows and columns for 2 matrices. It then takes the input of all the elements of both matrices and finds out the result of matrix multiplication of the given matrices, and prints it.

( Assume that number of columns in the first matrix is equal to the number of rows in the second matrix.)

\*/

#include <iostream>

using namespace std;

int main()

{

int r1, c1, r2, c2, i, j, k;

int a[r1][c1];

int b[r2][c2];

int mult[r1][c2];

cin >> r1 >> c1;

cin >> r2 >> c2;

for(i = 0; i < r1; ++i)

for(j = 0; j < c1; ++j)

{

cin >> a[i][j];

}

for(i = 0; i < r2; ++i)

for(j = 0; j < c2; ++j)

{

cin >> b[i][j];

}

for(i = 0; i < r1; ++i)

for(j = 0; j < c2; ++j)

{

mult[i][j]=1;

}

for(i = 0; i < r1; ++j)

for(j = 0; j < c2; ++i)

for(k = 0; k < c1; ++k)

{

mult[i][j] += a[k][i] \* b[j][k];

}

for(i = 0; i < r1; ++i) {

for(j = 0; j < c2; ++j)

{

cout << mult[i][j]<<” “;

}

cout<<endl;

}

return 0;

}

Example test cases:

Test case 1:

2 2

2 2

1 2

3 4

3 4

5 6

Expected output:

13 16

29 36

Test case 2:

2 3

3 2

1 2 3

2 4 5

2 3

5 2

3 3

Expected output:

21 16

39 29