Zevi Rubin

Test 1 corrections.

0: a:

#include <iostream>

using namespace std;

int main()

{

for (int a = 1000; a > 0; a--){

cout << a << endl;

}

cout << "Blastoff" << endl;

return 0;

}

b:

#include <iostream>

using namespace std;

int main()

{

int count = 1000;

while (count > 0){

cout << count << endl;

count--;

}

cout << "Blastoff" << endl;

return 0;

}

c:

#include <iostream>

using namespace std;

int main()

{

int count = 1000;

do

{

cout << count << endl;

count--;

} while (count > 0);

return 0;

}

1:

#include <iostream>

using namespace std;

int main()

{

char LETTER;

if (LETTER == A)

cout << "Alpha" << endl;

else if (LETTER == B)

cout << "Beta" << endl;

else if (LETTER == C)

cout << "Gamma" << endl;

else

cout << "Huh?" << endl;

return 0;

}

2:

#include <iostream>

using namespace std;

int main()

{

char LETTER = '';

switch (LETTER)

{

case 'A':

cout << "Alpha" << endl;

break;

case 'B':

cout << "Beta" << endl;

break;

case 'C':

cout << "Gamma" << endl;

break;

default:

cout << "huh" << endl;

}

return 0;

}

3:

A:

#include <iostream>

using namespace std;

int main()

{

double A[20000];

double hmr = 0;

return 0;

}

B:

20,000\*8 = **160000**

20,000 doubles, at 8 bytes each.

C:

int INIT(double A[], short& hmr)

{

int i=0;

while (i < MAX)

{

A[i] = 0;

i++;

}

return 0;

}

D:

#include <iostream>

using namespace std;

int INIT(double A[], short& hmr);

int getData(double A[], short& hmr);

const long MAX = 20000;

int main()

{

double A[MAX];

short hmr = 0;

INIT(A, hmr);

getData(A, hmr);

return 0;

}

int INIT(double A[], short& hmr)

{

int i=0;

while (i < MAX)

{

A[i] = 0;

i++;

}

return 0;

}

int getData(double A[], short& hmr)

{

int input;

cout << "Enter a number: ";

cin >> input;

while (input >= 0)

{

A[hmr] = input;

hmr++;

cout << "Enter another number: ";

cin >> input;

}

return 0;

}

E:

#include <iostream>

using namespace std;

int INIT(double A[], short& hmr);

int getData(double A[], short& hmr);

int printData(double A[], short hmr);

const long MAX = 20000;

int main()

{

double A[MAX];

short hmr = 0;

INIT(A, hmr);

getData(A, hmr);

printData(A, hmr);

return 0;

}

int INIT(double A[], short& hmr)

{

int i=0;

while (i < MAX)

{

A[i] = 0;

i++;

}

return 0;

}

int getData(double A[], short& hmr)

{

int input;

cout << "Enter a number: ";

cin >> input;

while (input >= 0)

{

A[hmr] = input;

hmr++;

cout << "Enter another number: ";

cin >> input;

}

return 0;

}

int printData(double A[], short hmr)

{

int i = 0;

while (i < hmr)

{

double output = A[i];

cout << output << ", " << endl;

i++;

}

cout << "DONE" << endl;

}

Question 4:

Unfortunately I was unable to get this to work either for my single struct filled with my info, or for the array of structs for the 10 students. I hope I am able to meet with you or Devon to get this sorted out.

This is what I did get though!

#include <iostream>

#include <string>

using namespace std;

int fill\_dataBase();

struct studentInfo

{

string Name;

short Years;

char ManWoman;

short Number;

};

studentInfo allRecords[100]; //struct name

studentInfo myInfo[1];

// struct studentInfo = foobar;

int main()

{

int i = 0;

string Name;

short Years;

char ManWoman;

short Number;

cout << "Enter your first name: ";

cin >> Name;

cout << "Enter your age: ";

cin >> Years;

cout << "Enter your gender (m/f): ";

cin >> ManWoman;

cout << "Enter your ssn: ";

cin >> Number;

// myInfo[0] = foobar;

// foobar.Name = Name;

// foobar.Years = Years;

// foobar.ManWoman = ManWoman;

// foobar.Number = Number;

myInfo[i].Name = Name;

myInfo[1].Years = Years;

myInfo[1].ManWoman = ManWoman;

myInfo[1].Number = Number;

//printf("\nYour name is: ", myInfo[i].Name);

printf("\nYour age is: ", myInfo[0].Years);

printf("\nYour gender is: ", myInfo[0].ManWoman);

printf("\nYour ssn is: ", myInfo[0].Number);

// printf("\nYour age is: ", foobar[0].Years);

// printf("\nYour gender is: ", foobar[0].ManWoman);

// printf("\nYour ssn is: ", foobar[0].Number);

fill\_dataBase();

}

int fill\_dataBase()

{

int i = 0;

string Name;

short Years;

char ManWoman;

short Number;

cout << "Enter name: ";

cin >> Name;

while (Name != "Done")

{

cout << "Enter age: ";

cin >> Years;

cout << "Enter gender: ";

cin >> ManWoman;

cout << "Enter ssn: ";

cin >> Number;

allRecords[i].Name = Name;

allRecords[i].Years = Years;

allRecords[i].ManWoman = ManWoman;

allRecords[i].Number = Number;

i++;

cout << "Enter name: ";

cin >> Name;

}

cout << "DONE" << endl;

}