CSE141 Introduction to Programming (Fall'23)

Sample Midterm Examination

Max Marks: 70 *Time Allowed:* 2 hours

Name: _____ **ERP:** _____

Question:	Tracing	Statements	Short Answers	Coding	Total
Marks:	15	10	15	30	70
Score:					

Question 1: Tracing ..... *15 marks*

Find the output of the following programs.

(a)

```
#include <iostream>
using namespace std;
int main()
{
  int grade = 100;
  if (grade > 60 && grade < 100)
    cout << "Pass";
  else
    cout << "Fail";
}
```

Solution: `Fail`

(b)

```
#include <iostream>
using namespace std;
int main()
{
  int i = 85, j = 2, k = 4;
  i--;
  j++;
  k -= 6;
  i = i / j + i / k;
  cout << i;
}
```

Solution: `-14`

(c)

```
#include <iostream>
using namespace std;
int main()
{
  int a = 14, b = -15, c = 22, d = 11;

  if(a > 12 && b < -14 && c < d)
    cout << "YAY!!!!";
  else if (d - 20 < b)
    cout << "GO DO MORE PROGRAMMING!";
  else if(a + 12 >= c)
    cout << "This is vexing......";
  else
    cout << "This is very fun!";
}
```

Solution: This is vexing......

(d)

```
#include<iostream>
#include<cmath>
int main() {
  int N = 5;
  int vec[N] = {8,-2,4,5,-8};

  for(int i = 0; i < N; i++) {
    int y = abs(vec[i]);
    for(int j = 0; j < y; j++) {
      std::cout << vec[i] << + " ";
    }
    std::cout << "\n";
  }
}
```

Solution:

```
8 8 8 8 8 8 8 8
-2 -2
4 4 4 4
5 5 5 5 5
-8 -8 -8 -8 -8 -8 -8 -8
```

(e)

```
#include <iostream>
int main()
{
  int arr[]  = {4, 2, 3, 0, 1};
  int count = 0;
  int value = arr[0];
  while (arr[value] > 0) {
    count++;
    arr[value] = arr[ arr[value] ];
  }
  std::cout << count;
}
```

Solution: 3

Question 2: Statements ..... *10 marks*

(a) [3 marks] Find errors, if any, in the following C++ statements. Assume all variables are declared and initialized.

(i) `cout << "x="x;`

(ii) `m = 5; //n = 10; // = m + n:`

(iii) `cin >> x; >> y;`

(iv) `cout << "Enter value:"; cin >> x;`

(v) `/*Addition*/z = x + y;`

(vi) `bool isA = (90 <= grade <= 100);`

Solution: (i) missing « before `x`
(ii) no error
(iii) there shoud not be a `;` after `cin » x`
(iv) no error
(v) mo error
(vi) `90 <= grade <= 100` is not a valid boolean expression. It should be `(90 <= grade) && (grade <= 100)`

(b) [1 mark] Which of the following statements are legal?

(i) `bool b = 1;`

(ii) `bool b = true;`

(iii) `bool b = "true";`

(iv) `bool b = True;`

Solution: (i): ok, `1` will be converted to `true`.
(ii): ok.
(iii): `"true"` is a string literal, which cannot be converted to a `bool` value.
(iv): `True` is not a valid `bool` value.

(c) [3 marks] Make changes in the following program so that it uses a `switch-case` statement in place of the nested `if-else` statement.

```
#include <iostream>
using namespace std;
int main()
{
  int legs=0;
  cout << "Enter number of legs ";
  cin >> legs;
  if(legs == 1) cout << "Pirate";
  else if(legs == 2) cout << "Bird";
  else if(legs == 4) cout << "Cat";
  else if(legs == 6) cout << "Spider";
  else cout << "I don't know what you are";
}
```

Solution:

```
switch(legs) {
  case 1: cout << "Pirate"; break;
  case 2: cout << "Bird"; break;
  case 4: cout << "Cat"; break;
  case 6: cout << "Spider"; break;
  default: cout << "I don't know what
      you are";
}
```

(d) [3 marks] Make changes in the following program so that it uses a `while` loop in place of the `for` loop.

```
#include <iostream>
using namespace std;
int main()
{
  int max;
  cout << "Enter max number";
  cin >> max;
  for(int i=max; i>=0; i--)
    cout << i << "\n";
}
```

Solution:

```
#include <iostream>
using namespace std;
int main()
{
  int i, max;
  cout << "Enter max number";
  cin >> max;
  i=max;
  while (i>=0) {
    cout << i << "\n";
    i--;
  }
}
```

Question 3: Short Answers ..... *15 marks*

(a) [2 marks] Suppose that `b[]` is an array of `100` elements, with all entries initialized to `0`, and that `a[]` is an array of `N` elements, each of which is an integer between `0` and `99`. What is the effect of the following loop?

```
for (j = 0; j < N; j++)
  b[a[j]]++;
```

Solution: It counts the number of occurrences of each integer between `0` and `99` in `a[]`. After the loop, `b[i]` contains the number of occurrences of `i` in `a[]`.

(b) [3 marks] Which of the following require using arrays. For each, the input comes from standard input and consists of `N` real numbers between `0.0` and `1.0`.

(i) Print the maximum element.

(ii) Print the maximum and minimum elements.

(iii) Print the median element.

(iv) Print the element that occurs most frequently.

(v) Print the sum of the squares of the elements.

(vi) Print the average of the N elements.

(vii) Print the element closest to 0.

(viii) Print all the numbers greater than the average.

(ix) Print the N elements in increasing order.

(x) Print the N elements in random order.

(xi) Print histogram (with, say 10 bins of size 0.1).

Solution: Only (iii), (iv), (viii), (ix), and (x) require using arrays. Other problems can be solved using a single variable to store the current best solution.

(c) [2 marks] Rank the order in which arithmetic statements are evaluated (place numbers 1, 2, ... in the space provided. If a rule is given that is not a correct rule, do not place a number next to it.

_____ exponentiation

__3__addition and subtraction, left to right

_____ addition and subtraction, right to left

__1__sub-expressions in parenthesis

__2__multiplication and division, left to right

_____ multiplication and division, right to left

(d) [2 marks] Suppose your friend has the following lines of code that intend to find the first index of the first positive integer in `array[0] ... array[N-1]`, where `array` is an array of `N` integers

```
int i = 0;
while (array[i] >=0 ) {
  i++;
}
location = i;
```

Will your friend's code work as intended?

Solution: When array contains at least one negative integer

(e) [3 marks] What does the following code-fragment does?

```
int n=26375, r=0;
while(n!=0) {
   int rem = n%10;
   r = r*10 + rem;
   n /= 10;
}
cout << r << endl;
```

Solution: It reverses the digits of `n`.

(f) [3 marks] What does the code between `// START` and `// END` do in the following program?

```
#include <iostream>
using namespace std;

int main()
{
  int N=10, val = 8;
  int arr[N] = {7,2,6,4,3,6,8,5,4,3};

  bool found = false;
  int i;
  for(i = 0; i < N && !found; i++) {
    if(arr[i] == val)
      found = true;
  }

  // START
  if(found) {
    for(int j = i; j < N; j++) {
      arr[j - 1] = arr[j];
    }
    arr[N - 1] = 0;
  }
  // END

  for(i = 0; i < N; i++)
    cout << arr[i] << " ";

}
```

Solution: It removes the first occurrence of `val` from `arr` and shifts the remaining elements to the left.

Question 4: Coding ..... *30 marks*

(a) [10 marks] A ball is thrown up in the air. Its height above the ground is given by the formula $h = 3 + 10t - t^2$ where t is the time in seconds after it is thrown. For example, at

Time t = 0, h = 3 + 0 - 0 = 3

Time t = 2, h = 3 + 20 - 4 = 19

When the ball hits the ground, h becomes negative. Write a fragment of C++ code to print out the height, h, once every second, starting a $t = 0$, and continuing until h becomes negative (meaning the ball has hit the ground). (Do not print the negative value.) Include all necessary data declarations and initializations.

Solution:

```
int t = 0;    //start at time 0
int h = 3;    //the initial value for h when t is 0
while ( h >= 0 ) {
  cout << "Height: " << h << endl;
  t = t + 1;
  h = 3 + ( 10 * t ) - ( t * t );
}
```

(b) [10 marks] In many cases when we want to find the average of a set of numbers, we throw away the extreme values, either because they

- are not representative (i.e. a 0 on a quiz because the person was absent)
- are presumed to be errors in measurement (in a bunch of 80 degree temperatures there is a 180 degree reading – someone used a magnifier to focus the sun's rays)

Write a complete program to read a series of numbers from the keyboard. Use an end-of-input signal value of −99. Calculate and print the average of the numbers between 32 and 212. Any number below 32 or above 212 should not be included in the average calculations, but you should keep a count of them and print that at the end of the program. A run of the program should resemble the following:

```
Enter a number (-99 to quit): 22
Enter a number (-99 to quit): 44
...
Enter a number (-99 to quit): -99

Average is: 45.45

There were 3 extreme values
```

Solution:

```
#include <iostream>
#include <iomanip>
using namespace std;

int main()
{
  const int LOW = 32;
  const int HIGH = 212;
  const int EXIT = -99;

  int userVal, betweenCnt = 0, extremeCnt = 0, sum = 0;
  double avg;

  //get the first number from the user
  cout << "Enter a number (" << EXIT << " to quit):";
  cin >> userVal;

  //while the user has not entered the exit value
  while( userVal != EXIT ) {
    // if the user's value is an extreme, then increment the count of
    // extreme values. Otherwise, add the user's value to the running
    // total and increment the count of the valid values

    if( userVal < LOW || userVal > HIGH )
      extremeCnt++;
```

```
    else {
      sum += userVal;
      betweenCnt++;
    }


    //get the next number from the user
    cout << "Enter a number (" << EXIT << " to quit):";
    cin >> userVal;
  }

  // If the user entered at least 1 valid number, calculate the average
  // and then display it and the number of extreme values that were
  // entered.

  if ( betweenCnt > 0 ) {
    avg = (double) sum / betweenCnt;
    cout << endl << endl << "Average is: "
         << fixed << setprecision(2) << avg
         << endl << "There were " << extremeCnt << " extreme values";
  }

  return 0;
}
```

(c) [10 marks] Write a program-fragment that prints out the sum of each row and each column of a given a N-by-N array `mat`. For example, with the following matrix:

```
const int N = 3;
int mat[N][N] = {{1, 2, 3},
                 {4, 5, 6},
                 {7, 8, 9} };
```

the output should be:

```
Row sums: 6 15 24
Column sums: 12 15 18
```

Solution:

```
#include <iostream>
using namespace std;

int main()
{
  const int N = 3;
  int mat[N][N] = {{1, 2, 3},
                   {4, 5, 6},
                   {7, 8, 9} };

  cout << "Row sums: ";
  for(int i=0; i<N; i++) {
    int sum = 0;
    for(int j=0; j<N; j++)
      sum += mat[i][j];
    cout << sum << " ";
  }

  cout << "\nColumn sums: ";
  for(int j=0; j<N; j++) {
    int sum = 0;
    for(int i=0; i<N; i++)
      sum += mat[i][j];
    cout << sum << " ";
  }

}
```