CS101 EndSem Practice

Spring 2023-24

Dhriti Maniar and Harsh Chaurasia

Find all the errors (the sum function prints the last element of s and (0,0) element of q):

```
#include <iostream>
using namespace std;
int sum(int q[][], int s, int len)
{
   int sum;
   sum = q[0][0] + s[len];
   return sum;
}
```

```
int main()
{
    int a[] = {7,90};
    int len = sizeof(a);
    int b[5] = {9,16};
    int p[][2] = {{12,2},{3,78}};
    int m = sum(p, a, len);
    cout<<m;
}</pre>
```

nt q[][2]

the no. of columns must always be passed for 2d array

```
#include <iostream>
using namespace std;
int sum(int q[]], int s, int len)
{
  int sum;
  sum = q[0][0] + s[len];
  return sum;
}
```

If len is the length of the array then make sure to access the last element with the index (len-1).

int ell

Do not forget this bracket while passing!

int len = sizeof(a)/sizeof(a[0]

Just using sizeof(a) will give a value of 8, which is the total space taken by the array.

```
int main()
{
  int a[] = {7,90};
  int len = sizeof(a);
  int b[5] = {9,16};
  int p[][2] = \{{12,2},{3,78}\};
  int m = sum(p, a, len);
  cout<<m;
}</pre>
```

This is correct. The other elements of array b will be initialized to 0 by default. b = {9,16,0,0,0}
But be careful, only declaring int b[5], stores arbitrary values in b.

What is happening in this function? Give a brief description.

Is the function returning any value or making changes to the original array passed from the main function?

```
void new function(int arr[], int n)
  int i, j;
  bool condition;
  for (i = 0; i < n - 1; i++)
     condition = false;
     for (j = 0; j < n - i - 1; j + +) {
        if (arr[j] > arr[j + 1]) {
           int t = arr[i];
           arr[i] = arr[i+1];
           arr[j+1] = t;
           condition = true;
     if (condition == false)
        break;
```

Pass by reference Conditional statements

Solution 2

This function is a sorting algorithm, namely Bubble sort. It sorts a given array in ascending order. For every integer it checks it if it larger than the next element. If no element is larger than its successor, it means that the array is sorted and it exits the current loop.

This function does not return any value. The original array which is being passed gets sorted. This is because arrays are always passed by reference.

```
void new function(int arr[], int n)
  int i, j;
  bool condition:
  for (i = 0; i < n - 1; i++)
     condition = false;
     for (j = 0; j < n - i - 1; j++)
        if (arr[j] > arr[j + 1]) {
           int t = arr[i];
           arr[i] = arr[i+1];
           arr[i+1] = t;
           condition = true;
     if (condition == false)
        break;
```

The function check_parantheses accepts a string which is a sequence of brackets. If the sequence is correct, this function prints valid and if it is not then it prints invalid.

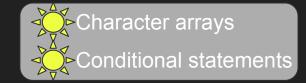
Assume that always a combination of '(' and ')' is passed.

Eg.

()()(()()) prints valid

)()()(prints invalid

```
void check parantheses(char s[])
  int c = 0, len = 0;
  while(
     len++:
  for (int i=0; i<len;i++){
     if (s[i]=='(')
        C++:
     else if (s[i]==')')
        cout<<"invalid";
        return;
     cout<<"valid";
  else
     cout<<"invalid";
```

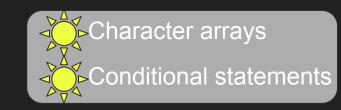


A char string has the last position always occupied by a token which is '\0'.

The value of c should never be less than 0, as this indicates there are more closing brackets.

The final value of c should be 0 if the opening and closing brackets are balanced

```
void check parantheses(char s[])
  int c = 0, len = 0;
  while(s[len]!='\0')
     len++:
   for (int i=0; i<len;i++){
     if (s[i]=='(')
        C++:
     else if (s[i]==')')
        C--;
     if(<u>c<0</u>){
        cout<<"invalid";
        return;
   if(c==0)
     cout<<"valid";
   else
     cout<<"invalid";
```



```
void alphabetical(char s1[], char s2[])
  int len1 = find length(str2);
  int len2 = find length(str2);
  int minLength = (len1 < len2) ? len1 : len2; // Get the minimum length
  int i = 0:
  while (i < minLength && str1[i] == str2[i]){
                   // Move to the next character if they are equal
     j++:
  if (i == len1 && i == len2) {
     cout << "Both strings are equal.";</pre>
  } else if ( i == len1 || str1[i] < str2[i] ){</pre>
     cout << str1 << " comes before " << str2 << " alphabetically.";
  } else {
     cout << str2 << " comes before " << str1 << " alphabetically.";
  return 0;
```

What will be the output of this program?

How many times is the function t being called?

```
# include <iostream>
using namespace std;
void t(int n, char fp , char tp , char ap) {
  if(n == 1) {
   cout << fp << tp;
  return;
  t(n -1,fp,ap, tp);
  cout << fp << tp;
  t(n -1,ap,tp,fp);
  return;
int main () {
  t(2, 'x','y','z');
  return 0;
```



Output:

XZXYZY

No. of times t is called:

3

Is this similar to the Towers of Hanoi problem?

```
# include <iostream>
using namespace std;
void t(int n, char fp , char tp , char ap) {
  if(n == 1) {
  cout << fp << tp;
  return;
  t(n -1,fp,ap, tp);
  cout << fp << tp;
  t(n -1, ap, tp, fp);
  return;
int main () {
  t(2, 'x','y','z');
  return 0;
```

If a = 5 and b = 3, what does this function return?

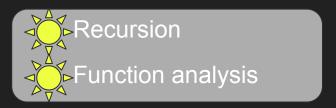
Give a brief description of what this function does.

```
int mystery (int a, int b) {
    if(a == 0)
    return 0;
    int temp = mystery (a - 1, b);
    if( temp == (b - 1))
    return 0;
    else
    return temp + 1;
}
```

Output:

2

This function finds a%b



```
int mystery (int a, int b) {
    if(a == 0)
    return 0;
    int temp = mystery (a - 1, b);
    if( temp == (b - 1))
    return 0;
    else
    return temp + 1;
}
```

Given the following template, complete the code for a structure with three floating-point variables: x, y, and z, and another new structure that can store three floating-point variables, r, θ , and ϕ (the spherical polar coordinate system).

You now need to write a member function for the structure Polar that can convert a Polar point to a Cartesian one.

```
struct Cartesian{
   float x, y, z;
};
struct Polar{
   float r, theta, phi;
      Define the function here
   . . . . . . (. . . . ) {
};
```

- Create a member function with an appropriate name
- Keep the return type as Cartesian
- Be mindful of using the correct convention. As we are using member variables in this member function, we write theta and not <object_name>.theta
- But wait, what about the main function?

```
struct Cartesian{
   float x, y, z;
                            In this case, we do not
};
                               need any other
struct Polar{
                             parameters. Just the
                            member variables will
   float r, theta, phi;
                                     do!
   Cartesian to Cartesian() {
       Cartesian ans;
       ans.x = r * sin(theta) * cos(phi);
       ans.y = r * sin(theta) * sin(phi);
       ans.z = r * cos(theta);
       return ans;
                                Returns a
                            Cartesian structure
};
                                 variable
```

Solution 6(contd.)

 Keep in mind how to use the structure variables and how to call the member functions too!

```
int main() {
   Polar p;
   cin >> p.r >> p.theta >> p.phi;
   Cartesian c_point = p.to_Cartesian();

   cout << c_point.x << endl << c_point.y << endl << c_point.y << endl << c_point.y;
}</pre>
```

Question 7: Find all the errors (both compilation and logical) and correct them!

```
struct Point{
  int x:
   int y;
class Circle{
  Point center;
   int radius;
  Circle constructor(int a, int b, int r){
     a = center.x;
     b = center.y;
     radius = r:
```

```
int main(){
  int a, b, r;
  cin >> a >> b >> r:
  Circle C1(a, b, r);
  // Print all the member variables in C1
  cout << "The center coordinates are: (" << C1.x << "," << C1.y << ")\n";
  cout << "The radius is: " << C1.radius:
```

```
struct Point{
                       Missing
  int x:
                     semi-colon
  int y;
class Circle{
  Point center;
  int radius;
  Circle constructor(int a, int b, int r){
     a = center.x:
     b = center.y;
     radius = r;
```

```
int main(){
  int a, b, r;
  cin >> a >> b >> r;
  Circle C1(a, b, r);
  // Print all the member variables in C1
  cout << "The center coordinates are: (" << C1.x << "," << C1.y << ")\n";
  cout << "The radius is: " << C1.radius;
```

```
struct Point{
                          Missing
  int x:
                        semi-colon
   int y;
                          A class is private by
class Circle{
                         default. So, you can't
                           use its member
   Point center;
                         variables outside the
   int radius;
                            class definition
  Circle constructor(int a, int b, int r){
     a = center.x:
     b = center.y;
     radius = r;
```

```
int main(){
  int a, b, r;
  cin >> a >> b >> r;
  Circle C1(a, b, r);
  // Print all the member variables in C1
  cout << "The center coordinates are: (" << C1.x << "," << C1.y << ")\n";
  cout << "The radius is: " << C1.radius:
```

```
struct Point{
                                             int main(){
                        Missing
  int x:
                                                int a, b, r;
                      semi-colon
   int y;
                                                cin >> a >> b >> r;
                                                Circle C1(a, b, r);
                       A class is private by
class Circle{
                       default. So, you can't
                         use its member
                                                // Print all the member variables in C1
  Point center;
                       variables outside the
                                                cout << "The center coordinates are: (" << C1.x << "," << C1.y << ")\n";
   int radius;
                         class definition
                                                cout << "The radius is: " << C1.radius:
  Circle constructor(int a, int b, int r){
     a = center.x:
     b = center.y;
     radius = r;
                                             A constructor's
                                           name is the same
                                            as its structure's
```

```
struct Point{
                                            int main(){
                       Missing
  int x:
                                               int a, b, r;
                      semi-colon
   int y;
                                               cin >> a >> b >> r;
                                               Circle C1(a, b, r);
                       A class is private by
class Circle{
                      default. So, you can't
                         use its member
                                               // Print all the member variables in C1
  Point center;
                       variables outside the
                                               cout << "The center coordinates are: (" << C1.x << "," << C1.y << ")\n";
   int radius;
                         class definition
                                               cout << "The radius is: " << C1.radius;
  Circle constructor(int a, int b, int r){
     a = center.x:
     b = center.y;
     radius = r;
                   Assigning
                                            A constructor's
                                           name is the same
                 garbage value
                                            as its structure's
                 to meaningful
                    variables
```

```
int main(){
struct Point{
                       Missing
  int x:
                                               int a, b, r;
                     semi-colon
   int y;
                                               cin >> a >> b >> r;
                                               Circle C1(a, b, r);
                       A class is private by
class Circle{
                      default. So, you can't
                        use its member
                                              // Print all the member variables in C1
  Point center;
                      variables outside the
                                              cout << "The center coordinates are: (" << C1.x << "," << C1.y << ")\n";
   int radius;
                        class definition
                                              cout << "The radius is: " << C1.radius;
  Circle constructor(int a, int b, int r){
     a = center.x:
     b = center.y;
     radius = r;
                                                                                                   Not the correct way to
                                                                                                  access the variables x
                   Assigning
                                            A constructor's
                                                                                                            and y
                                          name is the same
                garbage value
                                           as its structure's
                 to meaningful
                    variables
```

Solution 7: Corrected code

```
struct Point{
  int x;
  int y;
class Circle{
  public:
         Point center;
         int radius;
  Circle(int a, int b, int r){
     center.x = a;
     center.y = b;
     radius = r;
```

```
int main(){
  int a, b, r;
  cin >> a >> b >> r;
  Circle C1(a, b, r);
  // Print all the member variables in C1
  cout << "The center coordinates are: (" << C1.center.x << "," <<
C1.center.y << ")\n";
  cout << "The radius is: " << C1.radius;
```

Question 8: Complete the following code (slide 1 of 2)

```
#include<iostream>
#include<fstream>
using namespace std;
struct Student{
  int r no:
  int grade;
  Student(){} // Constructor
int main(){
  ifstream
read from("raw.txt");
  ofstream
write into("database.txt");
  Student S[10]:
```

```
/* This loop should read data from the file raw.txt containing 20 space-separated values (ten pairs of roll numbers and grades) and store it in the array of structure variables. For example, if raw.txt contains:

20 9 21 7 . . .
```

Then, the roll number of the 0-th student is 20, and the corresponding grade is 9. Similarly, the roll number of the 1st student is 21; the corresponding grade is 7, and so on. */

```
for(int i = 0; i < 10; i++){
    int r, g;
    .... >> r >> g// Read values from raw.txt into r and g
    .......
    .....// Store the values into the corresponding structure variable
}
```

Question 8: Complete the following code (slide 2 of 2)

```
// This loop should store the data in the file database.txt.
  // The format should be:
  // The roll number of the student is: <roll number of the i-th student>
  // <newline>
  // The grade of the student is: <grade of the i-th student>
  // <newline>
  for(int i = 0; i < 10; i++){
   .....// For roll number
```

```
#include<fstream>
using namespace std;
struct Student{
  int r no;
  int grade;
 Student(){} // Constructor
int main(){
  ifstream
read from("raw.txt");
  ofstream
write into("database.txt");
  Student S[10];
```

/* This loop should read data from the file raw.txt containing 20 space-separated values (ten pairs of roll numbers and grades) and store it in the array of structure variables. For example, if raw.txt contains:

```
20 9 21 7 . . .
```

Then, the roll number of the 0-th student is 20, and the corresponding grade is 9. Similarly, the grade of the 1st student is 21; the corresponding grade is 7, and so on. */

```
for(int i = 0; i < 10; i++){
    int r, g;

    read_from >> r >> g// Read values from raw.txt into r and g
    S[i].r_no = r;
    S[i].grade = g; // Store the values into the corresponding structure
variable
```

```
// This loop should store the data in the file database.txt.
  // The format should be:
  // The roll number of the student is: <roll number of the i-th student>
  // <newline>
  // The grade of the student is: <grade of the i-th student>
  // <newline>
  for(int i = 0; i < 10; i++){
    write into << "The roll number of the student is: " << S[i].r no << endl; // For roll number
    write into << "The grade of the student is: " << S[i].grade << endl; // For grade
```

All the very best for your endsems!

All the very best for your endsems!

* and all your future endeavours :)