PDS Lab Section 15

December 17, 2020

Comments in C

- You can put comments in your C file
- These will not be looked at by the compiler, but makes your program much more readable
- We will start practicing it today
- C comment
  - Anything typed in between /*   and */

```
int main()
{
   int i;
   /* for loop to read numbers */
   for (i=0; i < 10; i=i+1)
   {
        printf(“%d  “, i);
   }
   printf(“\nValue of i at end of loop is %d\n”, i);
}
```

Header to put in your file

- Every program must start with a comment containing
  - Section No.
  - Roll No.
  - Name
  - Assignment No.
  - A one line description of the assignment
- Type the example header (replace with your name, roll no. assignment no. etc.) at the beginning of each of your C file, even before the #include <stdio.h>

Example Header

```
/*
*     Section 15
*     Roll No  :  20CS30010
*     Name     :  Your Name
*     Assignment No : 3
*     Description : Program to check points
*/
```

Naming your program

- Name your file with assgnX_Y.c, where X is the assignment number and Y is your roll no.
  - assgn3_20ME10010, assgn4_20CE30014,…..

Conditional Statement

if (expression) statement1 else statement2;

- Executes statement1 if expression evaluates to true, otherwise executes statement2
- May not have the else part

  if (expression) statement1;

- statement1 and statement2 can be any other statement, for ex.
  - Assignment statement
  - Another if statement
  - Another if-else statement
  - for statement
  - while statement
  - ….

```
int main()
{
   int a, b, c;
   scanf(“%d%d%d”, &a, &b, &c);
   if (a == b)
  {
        if (b == c)
                printf(“The numbers are equal\n”);
        else
                printf(“The numbers are not equal\n”);
  }
  else printf(“The numbers are not equal\n”);
}
```

for loop

```
for (a; b; c)
{
    ….;
    ….;
}
```

a: statement executed only once at beginning

b: condition checked at the beginning of each iteration

c: statement executed at the end of every iteration

Separate the parts a, b, c by semicolon, NOT comma

Example program 1

```
int main()
{
   int i;
   for (i = 0; i < 10; i = i+1)
  {
        printf(“%d  “, i);
  }
  printf(“\nValue of i at end of loop is %d\n”, i);
}
```

Prints the numbers 0 to 9 in the same line and then prints 10 in the next line with the message

Example program 2: sum of numbers

```
int main()
{
    int i, n, x, sum;
    scanf(“%d”, &n);
    sum = 0;
    for (i = 0; i < n; i = i+1)
   {
          scanf(“%d”, &x);
          sum = sum + x;
   }
   printf(“The sum of the numbers is %d\n”, sum);
}
```

Assignments

Assignment 3

- Read in the coordinates of three points (x1, y1), (x2, y2), (x3, y3)
  - Name the variables as x1, y1, x2, y2, x3, y3
  - Assume all coordinates are integers
- Check if the three points are collinear (lies on the same line)
- If yes,
  - Print a message saying “The points (..,..), (.., ..), (..,..) are collinear”, showing the three points

- If no (else part of the if statement),
  - Compute the lengths of the three sides of the triangle formed by the points and store them in variables named side1, side2, side3
  - Print the lengths of the three sides in a single printf statement with a message like “The lengths of the three sides are …”
  - Compute the area of the triangle formed and store in a variable called area. Print the area with a nice message like “The area of the triangle is …”
  - Print if the triangle formed is equilateral (all three sides have same length), isosceles (exactly two sides have same length), or neither equilateral nor isosceles (all three sides have different lengths)
    - So print messages like “The triangle formed is equilateral” or “ The triangle formed is isosceles” or “ The triangle formed is neither equilateral nor isosceles”

IMPORTANT thing to note

- In C, int/int division will truncate

```
int a = 5, b = 2;
float c;
c = a/b;
printf(“value of c is %f\n”, c);
```

You will not get 2.5, you will get 2

To get the correct value, you need to “convert” one of the values to float temporarily. Any one of the following will work

```
c = ((float) a)/b;     c = a/((float) b);     c = (a*1.0)/b;   c = 1/ (1.0*b);
```

- Temporary because a and b are still int type variables, it is float only for this operation.
- Note the brackets carefully

Assignment 4

Consider an integer X. Let Y be the set of all integers that divide X, including 1 but not including X. We want to check whether X is equal to the sum of the numbers in Y. For example, if X = 6, then Y = {1, 2, 3}, and 1 + 2 + 3 = 6. But if X = 12, then Y = {1, 2, 3, 4, 6} and 1 + 2 + 3 + 4 + 6 = 16 ≠ 12

- Read in an integer X
- Print if X is equal to the sum of the numbers in Y as defined above or not (print a message in either case)
  - Approach: In a for loop (till what?), find the numbers that divide X one by one starting from 1, and add to a running sum when you find one. After you come out of the for loop, check in an if-else statement for equality and print accordingly
- **Do NOT use arrays even if you know it. You will get 0 if you use arrays.**

Teaser Problem (not to be submitted)

- Read two positive integers A and B.
- Find the largest common digit in A and B and print it. In case there is no common digit, print "None".
- Do NOT use arrays even if you know it