C-Programming Practice

Part – 1

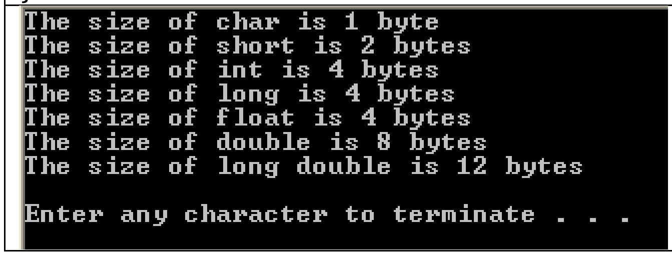
1. Data Types in C

The following table shows the basic data types in C:

Data Type	C Keyword	Bytes	Range	Placeholder (printf)	Placeholder (scanf)
Character	char	1	-128 to 127	%c	%c
Integer	int	4	-2,147,483,648 to 2,147,483,647	%d	%d
Floating Point	float	4	-3.4E38 to 3.4 E38	%f	%f
Double precession floating point.	double	8	-1.7e308 to 1.7e+308	%f	%lf

The following example will print the actual size allocated based on your computer:

```
/* Displays the number of bytes used to store each basic type */

#include <stdio.h>

int main(void) {
     printf("The size of char is %ld bytes\n", sizeof(char));
     printf("The size of short is %ld bytes\n", sizeof(short));
     printf("The size of int is %ld bytes\n", sizeof(int));
     printf("The size of long is %ld bytes\n", sizeof(long));
     printf("The size of float is %ld bytes\n", sizeof(float));
     printf("The size of double is %ld bytes\n", sizeof(double));
     printf("The size of long double is %ld bytes\n", sizeof(long double));
     return 0;
}
```



```
The size of char is 1 byte
The size of short is 2 bytes
The size of int is 4 bytes
The size of long is 4 bytes
The size of float is 4 bytes
The size of double is 8 bytes
The size of long double is 12 bytes

Enter any character to terminate . . .
```

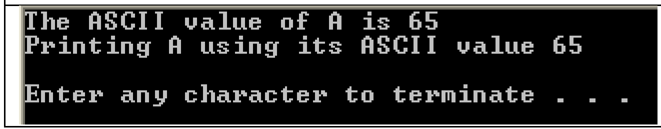
For compilation and execution on CentOS type

(1) $> `gcc file.c`, and (2) $> `./a.out` **OR** `$> gcc file.c -o file.out`, and `$> ./file.out`

Characters are actually represented in C as integer values. Each character is represented by its ASCII code (e.g A = 65. B = 66, etc). The table after the program below shows the printable ASCII characters and their corresponding ASCII codes.

Printing a char variable using “%c” will print the character but printing it with “%d” will print the ASCII code. Similarly, printing an integer variable with “%c” will also print the character provided the value is within the range of character values. The following example demonstrates this:

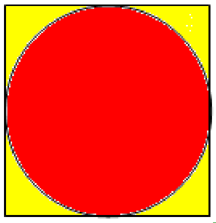
```
/* Shows the relationship between char and int types */
#include <stdio.h>

int main(void) {
     char c = 'A';
     int code = 65;
     printf("The ASCII value of %c is %d\n", c, c);
     printf("Printing %c using its ASCII value %d\n", code, code);
     return 0;
}
```

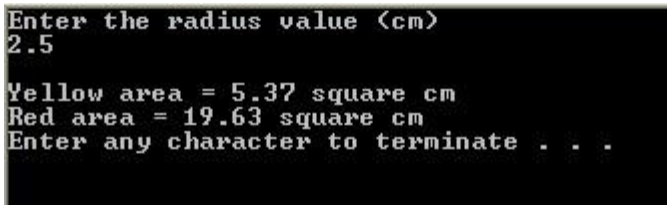


```
The ASCII value of A is 65
Printing A using its ASCII value 65

Enter any character to terminate . . .
```

Exercises:

1. Write an interactive C program that reads the radius of a circle inscribed inside a square and based on that it prints the values of red and yellow areas respectively. Assume that the radius is in cm.





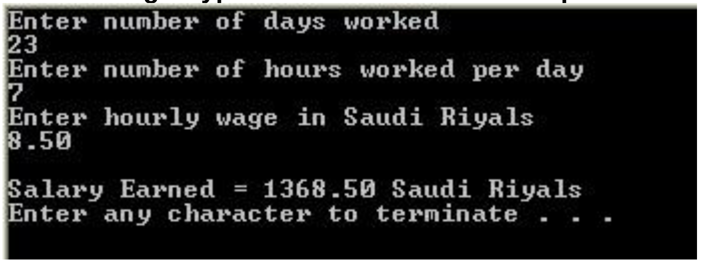
```
Enter the radius value (cm)
2.5

Yellow area = 5.37 square cm
Red area = 19.63 square cm
Enter any character to terminate . . .
```

Print each output value with two digits after the decimal point. Define the value of π (3.14159) as a constant. A sample run of your program gives the output above.

2. Write an interactive C program that computes the salary in Riyals of a part-time employee based on :

- **number of days worked.**
- **number of hours per day assuming that he worked the same hours every day.**
- **the hourly wage.**

Choose the right type for each variable. A sample run of the program is:



```
Enter number of days worked
23
Enter number of hours worked per day
7
Enter hourly wage in Saudi Riyals
8.50

Salary Earned = 1368.50 Saudi Riyals
Enter any character to terminate . . .
```

2. Arithmetic Operations in C

To solve most programming problems, you will need to write arithmetic expressions that manipulate data of type **int** and **double**. In C language, these are the basic arithmetic operators:

Arithmetic Operator	Operation
*	multiplication
/	division
%	remainder
+	Addition (binary +) or Unary +
-	Subtraction (binary -) or Unary -

Division between two integer numbers results in an integer result. Example 5/2 results in 2 and not 2.5. Division by zero is undefined. You get a compilation warning if you divide by the constant 0 and run-time error if you divide by a variable that has a value 0.

Standard Mathematical Functions

To do some advanced mathematical calculations, C provides a header file called **math.h** that has different trigonometric and algebraic functions. Here are some math functions defined in **math.h**:

C function	Mathematical Notation	Example
pow(x,y)	x^y	pow(5.0,3) = 5.0^3= 125.0
sqrt(x)	$\sqrt{x}$ ($x \geq 0$)	sqrt(4.0) = $\sqrt{4.0}$ = 2.0
log(x)	ln x ($x > 0$)	log(2.71828) = 1.0
log10(x)	$\log_{10}(x)$ ($x > 0$)	log10(100.0) = 2.0
exp(x)	e^x	exp(1.0) = 2.171828
sin(x)	sin x (x in radians)	sin(1.5708) = 1.0
cos(x)	cos x (x in radians)	cos(0.0) = 1.0
tan(x)	tan x (x in radians)	tan(0.0) = 0.0
asin(x)	$\sin^{-1} x$ ($x \in [-1,1]$)	asin (0) = 0
acos(x)	$\cos^{-1} x$ ($x \in [-1,1]$)	acos(0) = 1.570796
atan(x)	$\tan^{-1} x$	atan(0) = 0

Note:

- You must include the header file **math.h** before you use any of these functions.
- The return type of each of the above functions is **double**.

Example: Write a program that reads the lengths of two sides of a triangle (in cm) and the angle between them (in degrees) and calculates the length of the third side using the following formula:

$$side_3 = \sqrt{side_1{}^2 + side_2{}^2 - 2 \times side_1 \times side_2 \times \cos a}$$

```c
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

#define PI 3.14159

int main() {
  double side1, side2, side3, angle;   //use meaningful names for your
                                    //   variables
  printf("Enter side1 (cm): ");
  scanf("%lf", &side1);
  printf("Enter side2 (cm): ");
  scanf("%lf", &side2);
  printf("Enter angle in degrees: ");
  scanf("%lf", &angle);
  angle = angle/180 * PI;  // convert angle to radians
  side3 = sqrt(side1*side1 + side2*side2 - 2*side1*side2*cos(angle));
  //or side3=sqrt(pow(side1,2)+pow(side2,2)-2*side1*side2*cos(angle));
  printf("side3 = %.2f cm\n", side3);
  return 0;
}
```

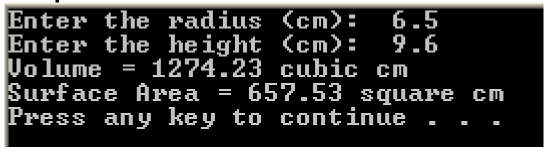
For compilation and execution on CentOS type

(1) $> `gcc file.c -lm`, and (2) $> `./a.out` OR `$>` `gcc file.c -o file.out -lm`, and `$>` `./file.out`

Exercises:

1. **Write a C program that prompts for and reads the radius and the height of a cylinder (in centimeters). It then computes its volume and surface area. The surface area is the surface of the two circular bases and the surface of the lateral side. Note: π = 3.14159. Here $V = \pi r^2 h$, $S = 2 (\pi r h + \pi r^2)$**

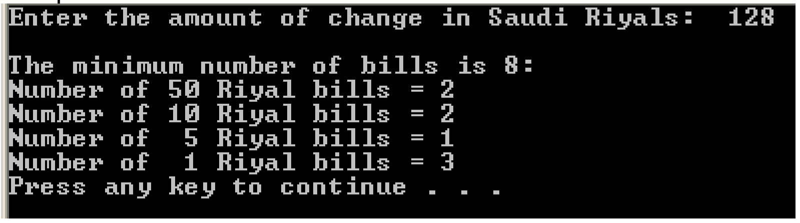
Sample run:



```
Enter the radius (cm):  6.5
Enter the height (cm):  9.6
Volume  = 1274.23 cubic cm
Surface Area = 657.53 square cm
Press any key to continue . . .
```

2. **Write a C program that prompts for and reads the amount of change in Saudi Riyals. It then finds and prints the minimum number of Saudi Riyal bills represented by the change. Assume that the bills available are 1, 5, 10, and 50 Riyals. Hint: Use remainder and integer division.**

Sample run:



```
Enter the amount of change in Saudi Riyals:  128

The minimum number of bills is 8:
Number of 50 Riyal bills = 2
Number of 10 Riyal bills = 2
Number of  5 Riyal bills = 1
Number of  1 Riyal bills = 3
Press any key to continue . . .
```

3. Selection Statements in C

The selection statements (if-else-if) and switch work very similar to the ones used in java. The only major difference is that there are no boolean variables in C. Instead, integers are used for boolean values in C.

Example (switch-statement):

```
#include <stdio.h>

int main(void) {
int num1, num2, result;
char operator;
printf("Enter two numbers and the operator\n");
scanf("%d %d %c", &num1, &num2, &operator);

switch(operator) {
   case '+': result = num1 + num2; printf("%d + %d = %d\n", num1, num2, result);
             break;
   case '-': result = num1 - num2; printf("%d - %d = %d\n", num1, num2, result);
             break;
   case '*': result = num1 * num2; printf("%d * %d = %d\n", num1, num2, result);
             break;
   case '/': if(num2 != 0) {
             result = num1 / num2; printf("%d / %d = %d\n", num1, num2, result);
             }
             else printf("Error!  num2 is zero\n");
             break;
   case '%': if(num2 != 0) {
             result = num1 % num2; printf("%d %% %d = %d\n", num1, num2, result);
             }
             else printf("Error!  num2 is zero\n");
             break;
    default: printf("Error:  Wrong operator");
    return 0; }
}
```

Example (if-statement):

```
#include <stdio.h>

int main(void) {
	int validGrade = 1;      //  set validGrade to true
	double grade; char letterGrade;
	printf("Enter grade> ");
	scanf("%lf", &grade);
	if(grade < 0.0 || grade > 100.0) validGrade = 0;    //  set validGrade to false
	else if(grade >= 85.0) letterGrade = 'A';
	else if(grade >= 75.0) letterGrade = 'B';
	else if(grade >= 65.0) letterGrade = 'C';
	else if(grade >= 45.0) letterGrade = 'D';
	else letterGrade = 'F';

	if(validGrade) printf("The letter grade is %c\n", letterGrade);
	else printf("Error: Invalid grade");

	return 0;
}
```

Exercises:

1. Write a program that prompts and reads two integer numbers. It then checks the numbers and prints one of the following messages accordingly:

You have entered two even numbers.
You have entered two odd numbers.
You have entered one even number and one odd number.

Hint: Use the modulus operator (%) for checking the numbers.

2. Use a *switch* statement to write a complete C program that prompts and reads an integer representing the number of days in a month (28, 29, 30, or 31). The program then prints the name of the month or months with that number of days. Your program must handle the case of wrong input within the switch statement.

You may find the following poem useful: *"Thirty days have September, April, June, and November. All the rest have thirty one, except February alone, that has twenty eight clear; but twenty nine in each leap year"*

4. Loops in C: for, while and do-while

Loops in C are similar to those in java. [One important difference is that you cannot initialize a variable in the body of a for-loop, i.e. for (int i = …) is not valid in C]

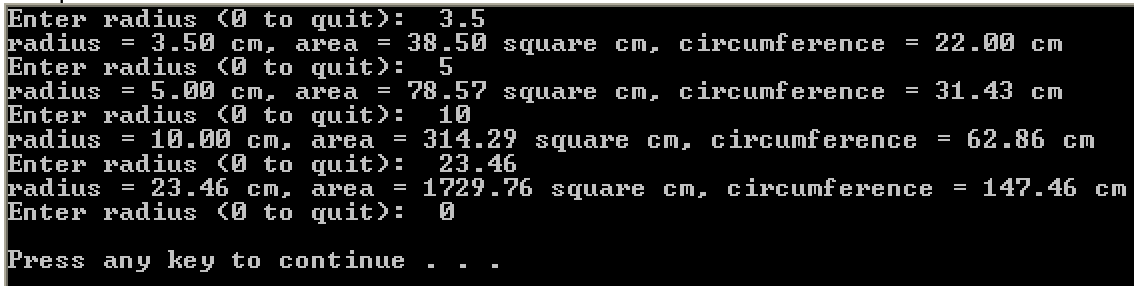
Example:

```
#include <stdio.h>
#include <stdlib.h>
#define NUMSTUDENTS 4
#define NUMQUIZES 3
int main(void){
      double grade, studentTotal, studentAverage;
      int m, n;

      for(m = 1; m <= NUMSTUDENTS; m++)
      {
          studentTotal = 0.0;
          for(n = 1; n <= NUMQUIZES; n++)
          {
             printf("Enter QuizGrade#%d for student#%d\n", n, m);
             scanf("%lf", &grade);
             studentTotal += grade;
          }
          studentAverage = studentTotal / NUMQUIZES;
          printf("The average for student#%d is %.2f\n", studentAverage);

      }

    return 0;
}
```

Exercises

1. Using sentinel controlled loop (**<u>while or for loop</u>**), write a program that reads a radius from the user and displays the area and circumference of the circle with that radius. This task is repeated until the user types 0. Assume that the radius is in centimeters. Use a named constant PI with value 3.1429

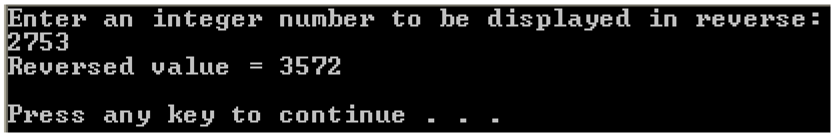
Sample run:



```
Enter radius (0 to quit):  3.5
radius = 3.50 cm, area = 38.50 square cm, circumference = 22.00 cm
Enter radius (0 to quit):  5
radius = 5.00 cm, area = 78.57 square cm, circumference = 31.43 cm
Enter radius (0 to quit):  10
radius = 10.00 cm, area = 314.29 square cm, circumference = 62.86 cm
Enter radius (0 to quit):  23.46
radius = 23.46 cm, area = 1729.76 square cm, circumference = 147.46 cm
Enter radius (0 to quit):  0

Press any key to continue . . .
```

2. Using a **<u>while loop</u>**, write a C program to display a positive integer number typed by the user in reverse.

Sample run:



```
Enter an integer number to be displayed in reverse:
2753
Reversed value = 3572

Press any key to continue . . .
```

Hint: Use remainder and division by 10.