1310 Practice Session 2020/10/24

This file is to be shared only with the students who attend 2020/10/24 practice session.

Bhanu Jain

Chapter 3

First C Programs

Q1 Class Exercise

- Write a program that does the following:
  - Prints 1 statement to output
  - Prints 1 statement to output
  - Declares a variable and assign value 0f 2.5 to it.
  - Prints 1 statement to output

```
1   2   3   4
```

```
1
2
3
4
```

```
x = 2.500000
```

Q1 Class Exercise

- Write a program that does the following:
  - Prints 1 statement to output
  - Prints 1 statement to output
  - Declares a variable and assign value 0f 2.5 to it.
  - Prints 1 statement to output

```
1   2   3   4
```

```
1
2
3
4
```

```
x = 2.500000
```

```
#include <stdio.h>
int main(){
   printf("1   2   3   4\n");
   printf("1   \n2   \n3   \n4  \n");
   double x=2.5;
   printf("x = %f\n",x);
 return 0;
}
```

Q2

- Write a program that does the following:
  - Prints “Have a nice day!”
  - Prints the sum of 6.3 + 12/7 .
  - Use the floating-point format specifier to print the result of the expression .

Q2

- Write a program that does the following:
  - Prints “Have a nice day!”
  - Prints the sum of 6.3 + 12/7 .
  - Use the floating-point format specifier to print the result of the expression .

```
#include <stdio.h>
int main()
{
    printf("Have a nice day!");
    printf ("%f", 6.3 + 12/7);
    return 0;
```

```
Have a nice day!7.300000
```

Q3

- Write a program that does the following:
  - Uses the floating−point format specifier to print the result precise to the **second decimal position** .
  - 2^{10}
  - $8\left(2+\frac{3.5}{7}\right)^4$
  - $\sqrt{3}$
  - $4-\sqrt{3+\frac{5}{7.2}}.$
  - $4+\sqrt{1+\frac{2}{1.2}}+\ 5\left(\frac{9}{7}\right)^3$
  - Produces output as shown here.

```
1024.00
312.50
1.73
2.08
10.63
```

Q3

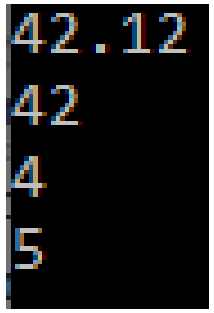
- Write a program that does the following:
  - Use the floating-point format specifier to print the result precise to the **second decimal position** .
  - 2^{10}
  - $8\left(2+\frac{3.5}{7}\right)^4$
  - $\sqrt{3}$
  - $4-\sqrt{3+\frac{5}{7.2}}$.
  - Produces o/p as shown here.

```
1024.00
312.50
1.73
2.08
10.63
```

```
#include <stdio.h>
#include <math.h>
int main()
{    printf("%0.2f\n",pow(2,10));
     printf("%0.2f\n",8 * pow(2 + 3.5/7, 4));
     printf("%0.2f\n",sqrt(3));
     printf("%0.2f\n",4 - sqrt(3+5/7.2));
     printf("%0.2f\n", 4+ sqrt(1+2/1.2) + 5* pow(9/7 , 3 ));
     return 0;
}
```

Q4

- Write a program that does the following:
  - Has two inline functions min(int a, int b) and max(int a, int b) that return min and max integers respectively
  - Prints floating point output precise to 2 decimal places.
  - Prints floating point absolute value of (-42.12) (Use **fabs()**)
  - Prints decimal absolute value of (-42) (Use **abs()**)
  - Prints min(5,4) using the **inline** min function
  - Prints max(5,4) using the **inline** min function
  - Produces an o/p as shown below:



Q4

- Write a program that does the following:
  - Has two inline functions min(int a, int b) and max(int a, int b) that return min and max integers respectively
  - Prints floating point output precise to 2 decimal places.
  - Prints floating point absolute value of (-42.12)
  - Prints decimal absolute value of (-42)
  - Prints min(5,4) using the inline min function
  - Prints max(5,4) using the inline min function
  - Produces an o/p as shown below

```
42.12
42
4
5
```

```
#include <stdio.h>
#include <math.h>
inline int max ( int a, int b ) { return a > b ? a : b;}
inline int min ( int a, int b ) { return a < b ? a : b;}
int main()
{
   printf ("%0.2f\n",fabs(-42.12));
   printf ("%d\n",abs(-42));
   printf ("%d\n",min(5, 4));
   printf ("%d\n",max(5, 4));
   return 0;
}
```

Q5

- Write a program that does the following:
  - Asks the user to enter the radius of a circle.
  - Computes the circumference of the circle.
  - Computes the area of the circle.
  - Prints the o/p as shown below:

```
Please enter the radius: 5.2
The circumference is 32.67
The area is 84.95
```

Q5

- Write a program that does the following:
  - Asks the user to enter the radius of a circle.
  - Computes the circumference of the circle.
  - Computes the area of the circle.
  - Prints the o/p as shown below:

```
Please enter the radius: 5.2
The circumference is 32.67
The area is 84.95
```

```c
#include <stdio.h>
#include <math.h>
int main()
{
    double radius;
    printf("Please enter the radius: ");
    scanf("%lf",&radius);
    double  circumference = 2 * M_PI * radius;
    double area = M_PI * pow(radius, 2);
    printf("The circumference is %0.2f\n", circumference);
    printf("The area is %0.2f\n", area);
    return 0;
}
```

Q6

- Write a program that does the following:
  - Declares a variable days and initializes it with 31
  - Declares a variable month of type character array and initializes it with "March"
  - Prints days variable and constant string "July" using appropriate format specifiers.
  - Prints the contents of the variable month and the ***expression*** *representing average of 85.1 and 85.5* using appropriate format specifiers.
  - Prints the o/p as shown below:

```
There are 31 days in July
Average temperature in July: 85.300 degrees
```

Q6

- Write a program that does the following:
  - Declares a variable days and initializes it with 31
  - Declares a variable month of type character array and initializes it with "March"
  - Prints days variable and **constant string** "July" using appropriate format specifiers.
  - Prints the contents of the variable month and the *expression representing average of 85.1 and 85.5* using **appropriate format specifiers**.
  - Prints the o/p as shown below:

```
There are 31 days in July
Average temperature in July: 85.300 degrees
```

```c
#include <stdio.h>
#include <math.h>
int main()
{
    int days = 31;
    char month[10] = "March";
    printf("There are %d days in %s\n", days, "July");
    printf("Average temperature in %s: %f degrees\n",
            month, (85.1 + 85.5) / 2.0);
 return 0;
}
```

Q7

- Write a program that does the following:
  - Asks the user to enter three numbers.
  - Computes and prints out the average of those numbers
  - Prints o/p as shown below:

```
Please enter the first number: 1
Please enter the second number: 2
Please enter the third number: 3
The average is 2.00
```

Q7

- Write a program that does the following:
  - Asks the user to enter three numbers.
  - Computes and prints out the average of those nur
  - Prints o/p as shown below:

```
Please enter the first number: 1
Please enter the second number: 2
Please enter the third number: 3
The average is 2.00
```

```c
#include <stdio.h>
#include <math.h>
int main()
{
    double n1, n2,n3;
    printf("Please enter the first number: ");
    scanf("%lf",&n1);
    printf("Please enter the second number: ");
    scanf("%lf",&n2);
    printf("Please enter the third number: ");
    scanf("%lf",&n3);
    double average = (n1 + n2 + n3) / 3.0;
    printf("The average is %0.2f\n", average);
    return 0;
}
```

Q8

- Write a program that does the following:
  - Asks the user to enter number of weeks.
  - Computes and print the number of days in “n” weeks
  - Prints o/p as shown below:

```
Enter number of weeks: 2
Result: 14 days
```

Q8

- Write a program that does the following:
  - Asks the user to enter number of weeks.
  - Computes and print the number of days in “n” weeks
  - Prints o/p as shown below:

```
Enter number of weeks: 2
Result: 14 days
```

```c
/* A program that converts weeks into days.*/
#include <stdio.h>
#include <math.h>
int main()
{
    int weeks =0;
    printf("Enter number of weeks: ");
    scanf("%d",&weeks); // Get user input

    int days = weeks * 7; // Converting weeks into days.
    printf("Result: %d days\n", days);
    return 0;
}
```

Chapter 4

Variables, Types, Operations on Numbers

Q1 Class Exercise

- Write a program that does the following:
  - Asks user to input a number **N** and receives it.
  - Computes Result = **N*** $\sqrt{\mathbf{1^2} + \boldsymbol{N^2}}$
  - Prints the Result, precise to 3 decimal places.

```
Please enter a number: 5.1
26.505
```

Q1 Class Exercise

- Write a program that does the following:
  - Asks user to input a number **N** and receives it.
  - Computes Result **N*** $\sqrt{\mathbf{1^2} + \boldsymbol{N^2}}$
  - Prints the result, precise to 3 decimal places.

```
Please enter a number: 5.1
26.505
```

```c
#include <stdio.h>
#include <math.h>
int main()
{
    double N, Result;
    printf("Please enter a number: ");
    scanf("%lf",&N);
    Result = N* sqrt(1+pow(N,2));
    printf("%0.3f\n",Result);
    return 0;
}
```

Sizes of Primitive Types

Depends on the architecture, but for our computers:

- byte: 8 bits
- short: 16 bits
- int: 32 bits
- long: 64 bits
- float: 32 bits
- double: 64 bits
- char: 8 bits

Given the data type of a variable, C knows how many bits to read from memory and how to interpret them.

The number of bits determines the **range of values** that can be represented using a given type.

Sizes of Primitive Types

Depends on the architecture, but for our computers:

Data Type	Memory(bytes)	Range	Format specifier
short int	2	-32,768 to 32,767	%hd
unsigned short int	2	0 to 65,535	%hu
unsigned int	4	0 to 4,294,967,295	%u
int	4	-2,147,483,648 to 2,147,483,647	%d
long int	8	-2,147,483,648 to 2,147,483,647	%ld
unsigned long int	8	0 to 4,294,967,295	%lu
long long int	8	-(2^63) to (2^63)-1	%lld
unsigned long long int	8	0 to 18,446,744,073,709,551,615	%llu
signed char	1	-128 to 127	%c
unsigned char	1	0 to 255	%c
float	4	1.2E-38 to 3.4E+38	%f
double	8	2.3E-308 to 1.7E+308	%lf
long double	16	3.4E-4932 to 1.1E+4932	%Lf

- Given the data type of a variable, C knows how many bits to read from memory and how to interpret them. The number of bytes determines the range of values that can be represented using a given type.

Q2

- Write the output of the following program without executing it. Then verify whether your results are correct

```
#include <stdio.h>
#include <math.h>
int main()
{
    double x = 5.5;
    x++;
    printf("%f\n",x);
    int y = 4;
    y--;
    printf("%d\n",y);
    return 0;
}
```

Q2

- Write the output of the following program without executing it. Then verify whether your results are correct

```
#include <stdio.h>
#include <math.h>
int main()
{
    double x = 5.5;
    x++;
    printf("%f\n",x);
    int y = 4;
    y--;
    printf("%d\n",y);
    return 0;
}
```

```
6.500000
3
```

Q3

- Write the output of the following program without executing it. Then verify whether your results are correct

```
#include <stdio.h>
#include <math.h>
int main()
{
    double x = 5.5;
    x += 3.2;
    int y = 20;
    y -= 5;
    printf("%f\n",x);
    printf("%d\n",y);
    return 0;
}
```

Q3

- Write the output of the following program without executing it. Then verify whether your results are correct

```
#include <stdio.h>
#include <math.h>
int main()
{
    double x = 5.5;
    x += 3.2;
    int y = 20;
    y -= 5;
    printf("%f\n",x);
    printf("%d\n",y);
    return 0;
}
```

```
8.700000
15
```

Q4

- Write a program that does the following:
  - Declares an integer variable weeks.
  - Declares a **constant** days_per_week of type integer and initialize it to 7 .
  - Asks user to enter number of weeks.
  - Computes and displays number of days in the given number of weeks.
  - Has an output as shown below

```
Please enter number of weeks:5
5 weeks = 35 days
```

Q4

```
Please enter number of weeks:5
5 weeks = 35 days
```

- Write a program that does the following:
  - Declares an integer variable weeks.
  - Declares a **constant** days_per_week of type integer and initialize it to 7 .
  - Asks user to enter number of weeks.
  - Computes and displays number of days in the given number of weeks.
  - Has an output as shown below

```
#include <stdio.h>
#include <math.h>
int main()
{   int weeks = 12;
    const int days_per_week = 7;
    int days = 0;
    printf("Please enter number of weeks:");
    scanf("%d",&weeks);
    days = weeks * days_per_week;
    printf("%d weeks = %d days\n", weeks, days);
    return 0;
}
```

Chapter 5

Formatted Output (printf)

Q1 Class Exercise

- Write a program that does the following:
  - Asks user to input a number and assigns it to **N**.
  - Asks user to input a number and assigns it to **M**.
  - Writes an **inline function max with parameters n and m. The function** returns the number that has a higher remainder when divided by 5.
  - Calls **max** with **arguments N** and **M**
  - Has an output as shown below

```
Enter N: 12
Enter M: 21
You entered 12 and 21.
The number that has a greater remainder when divided by 5 is: 12
```

Q1 Class Exercise

- Write a program that does the following:
  - Asks user to input a number and assigns it to **N**.
  - Asks user to input a number and assigns it to **M**.
  - Writes an **inline** **function max with parameters n and m. The function** returns the number that has a higher remainder when divided by 5.
  - Calls **max** with **arguments** **N** and **M**
  - Has an output as shown below

```
Enter N: 12
Enter M: 21
You entered 12 and 21.
The number that has a greater remainder when divided by 5 is: 12
```

```
#include <stdio.h>
inline int max(int n, int m) { return n%5 > m%5 ? n : m; }
int main()  {
   int N,M;
   printf("Enter N: ");
   scanf("%d", &N);
   printf("Enter M: ");
   scanf("%d", &M);
   printf("You entered %d and %d.\n",N,M);
   printf("The number that has a greater remainder when divided by 5 is: %d", max(N,M) );
  return 0;
  }
```

Q2

- Write a program that does the following:
  - Asks user to input a string.
  - Displays the string in reverse order.
  - Displays the numbers 1-30 as shown below.
  - Produces an output as shown below.

```
Enter String: Happy
 yppaH

1       2       3       4       5       6       7       10
11      12      13      14      15      16      17      20
21      22      23      24      25      26      27      30
```

Q2

- Write a program that does the following:
  - Asks user to input a string.
  - Displays the string in reverse order.
  - Displays the numbers 1-30 as shown below.
  - Produces an output as shown below.

```
Enter String: Happy
 yppaH

1       2       3       4       5       6       7       10
11      12      13      14      15      16      17      20
21      22      23      24      25      26      27      30
```

```
#include <stdio.h>
#include <string.h>
int main() {
   char S[20];
   printf("Enter String: ");
   scanf("%s",&S);
   for (int i=strlen(S); i >= 0; --i){
      printf("%c",S[i]);
   }
   printf("\n\n");
   for (int i=1; i<25;i++){
      printf("%o\t",i);
      if (i%8==0)     {printf("\n");}
   }
  return 0;
}
```

Q3

- Use the appropriate format specifiers so that:
  - Your program outputs as shown below:

```
#include <stdio.h>
int main(){
    int days = 31;
    char month[] = "July";
    printf("There are %? (octal)  days in %s\n", days, "July");
    printf("There are %? (hexadecimal)  days in %s\n", days, "July");
    printf("Average temperature in %s: %? degrees \n", month, (85.1 + 85.5) / 2.0 );
    return 0;
}
```

```
There are 37 (octal)  days in July
There are 1f (hexadecimal)  days in July
Average temperature in July: 8.530000e+001 degrees
```

Q3

- Use the appropriate format specifiers so that:
- Your program outputs

```
There are 37 (octal)  days in July
There are 1f (hexadecimal)  days in July
Average temperature in July: 8.530000e+001 degrees
```

```
#include <stdio.h>
int main(){
    int days = 31;
    char month[] = "July";
    printf("There are %o (octal)  days in %s\n", days, "July");
    printf("There are %x (hexadecimal)  days in %s\n", days, "July");
    printf("Average temperature in %s: %e degrees \n", month, (85.1 + 85.5) / 2.0 );
    return 0;
}
```

%e	floating point/double in **exponential format**
%o	**octa**l representation of an integer (base 8)
%x	**hexadecima**l representation of an integer (base 16)
%%	print a **percent sign**

Q4

- Use the appropriate format specifiers so that:
  - Your program outputs as shown below:

```
#include <stdio.h>
int main(){
    printf("%?s: %?f\n", "Dallas", 106.7431);
    printf("%?s: %?f\n", "San Francisco",  64.918262);
    printf("%?s: %?f\n", "surface of the sun",12000.0);
    return 0;
}
```

```
Dallas                :    106.74
San Francisco         :     64.92
surface of the sun    : 12000.00
```

```
                Dallas:    106.74
         San Francisco:     64.92
surface of the sun: 12000.00
```

Q4

- Use the appropriate format specifiers so that:
  - Your program outputs as shown below:

```
#include <stdio.h>
int main(){
    printf("%-20s: %8.2f\n", "Dallas", 106.7431);
    printf("%-20s: %8.2f\n", "San Francisco",  64.918262);
    printf("%-20s: %8.2f\n", "surface of the sun",12000.0);
    return 0;
}
```

```
#include <stdio.h>
int main(){
    printf("%20s: %8.2f\n", "Dallas", 106.7431);
    printf("%20s: %8.2f\n", "San Francisco",  64.918262);
    printf("%20s: %8.2f\n", "surface of the sun",12000.0);
    return 0;
}
```

```
Dallas               :   106.74
San Francisco        :    64.92
surface of the sun   : 12000.00
```

```
              Dallas:   106.74
       San Francisco:    64.92
surface of the sun: 12000.00
```

Q5

- Write a program that does the following:
  - Asks user to input a number.
  - Computes the square of the number.
  - Produces an output as shown below.

```
Enter a number: 5
The square of 5.00 is 25.00
```

```
Enter a number: 2.4
The square of 2.40 is 5.76
```

Q5

- Write a program that does the following:
  - Asks user to input a number.
  - Computes the square of the number.
  - Produces an output as shown below.

```
Enter a number: 5
The square of 5.00 is 25.00
```

```
Enter a number: 2.4
The square of 2.40 is 5.76
```

```c
#include <stdio.h>
#include <math.h>
int main(){
    double number;
    printf("Enter a number: ");

    scanf("%lf",&number);
    double square = pow(number, 2);
    printf("The square of %0.2f is %0.2f\n", number, square);
    return 0;
}
```

Q6

- What will be the outputs of the two programs shown below?

```
#include <stdio.h>
int main(){
    printf("1 %+d\n", 56);
    printf("2 %+5d\n", 56);
    printf("3 %5d\n", 56);
    printf("4 %05d\n", 56);
    printf("5 %07.2f\n", 56.15);
    printf("6 %+07.2f\n", 56.15);
    printf("7 Helloo\b World\n");
    printf("8 Hello\tWorld\n");
    printf("9 Hello\\World\n");
    return 0;
}
```

```
#include <stdio.h>
int main(){
    printf("1 %-d\n", 56);
    printf("2 %-5d\n", 56);
    printf("3 %5d\n", 56);
    printf("4 %05d\n", 56);
    printf("5 %07.2f\n", 56.15);
    printf("6 %-07.2f\n", 56.15);
    printf("7 Hellot\b World\n");
    printf("8 Hello\tWorld\n");
    printf("9 Hello\\World\n");
    return 0;
}
```

Q6

- What will be the outputs of the two programs shown below?

```
#include <stdio.h>
int main(){
    printf("1 %+d\n", 56);
    printf("2 %+5d\n", 56);
    printf("3 %5d\n", 56);
    printf("4 %05d\n", 56);
    printf("5 %07.2f\n", 56.15);
    printf("6 %+07.2f\n", 56.15);
    printf("7 Helloo\b World\n");
    printf("8 Hello\tWorld\n");
    printf("9 Hello\\World\n");
    return 0;
}
```

```
1 +56
2   +56
3    56
4 00056
5 0056.15
6 +056.15
7 Hello World
8 Hello World
9 Hello\World
```

```
#include <stdio.h>
int main(){
    printf("1 %-d\n", 56);
    printf("2 %-5d\n", 56);
    printf("3 %5d\n", 56);
    printf("4 %05d\n", 56);
    printf("5 %07.2f\n", 56.15);
    printf("6 %-07.2f\n", 56.15);
    printf("7 Hellot\b World\n");
    printf("8 Hello\tWorld\n");
    printf("9 Hello\\World\n");
    return 0;
}
```

```
1 56
2 56
3    56
4 00056
5 0056.15
6 56.15
7 Hello World
8 Hello World
9 Hello\World
```

Chapter 6

Strings

Q1 Class Exercise

- Write a program that does the following:
  - Asks the user to input a 3-word long sequence.
  - Uses gets(str1) instead of the scanf() function.
  - Prints the characters in the reverse order.
  - Produces an output as shown below.

```
Please enter a 3-word long sentence: I love mangoes!
You entered "I love mangoes!"
!seognam evol I
```

Q1 Class Exercise

- Write a program that does the following:
  - Asks the user to input a 3-word long sequence.
  - Uses gets(str1) instead of the scanf() function.
  - Prints the characters in the reverse order.
  - Produces an output as shown below.

```
Please enter a 3-word long sentence: I love mangoes!
You entered "I love mangoes!"
!seognam evol I
```

```
#include <stdio.h>
#include <math.h>
#include <string.h>
int main(){
   char str1[100];
   printf("Please enter a 3-word long sentence: ");
   gets(str1);
   printf("You entered \"%s\"\n",str1);

   for(int i=strlen(str1)-1;   i>=0;    i--){
      printf("%c",str1[i]);
   }

   return 0;
}
```

Q2 Class Exercise

- Write a program that does the following:
  - Asks the user to input a string.
  - Asks the user to input a number .
  - Prints the first N characters .
  - Prints the last N characters.
  - Produces an output as shown below:

```
Please enter a word: Golden Kiwi
Please enter a number: 4
You entered "Golden Kiwi" and a number "4"
The first 4 characters are: Gold

The last 4 characters are:   Kiwi
```

Q2 Class Exercise

- Write a program that does the following:
  - Asks the user to input a string.
  - Asks the user to input a number .
  - Prints the first N characters .
  - Prints the last N characters.
  - Produces an output as shown below:

```
Please enter a word: Golden Kiwi
Please enter a number: 4
You entered "Golden Kiwi" and a number "4"
The first 4 characters are: Gold

The last 4 characters are:   Kiwi
```

```
#include <stdio.h>
#include <string.h>
int main()
{   char str1[100];
    int N, len;
    printf("Please enter a word: ");
    gets(str1);   // scanf("%s", &str1);
    printf("Please enter a number: ");
    scanf("%d", &N);
    printf("You entered \"%s\" and a number \"%d\" \n", str1,N);
    printf("The first %d characters are: ",N);   //Needs to be outside the loop
    for(int i=0;  i  < N;  i++){
        printf("%c",str1[i] );    }
    len=strlen(str1);

    printf("\n\nThe last %d characters are:  %s",N, &str1[len-N] );
    return 0;
}
```

Q3

- Write a program that produces the following o/p:

Output:

```
Enter a string: birdie
you entered birdie
The number of characters in the
input string is 6.
After inserting a null character on
the third pos, length of string is 3.
Size of text (char array) is 100.
Size of int is 4 bytes.
Size of char is 1 byte.
Size of double is 8 bytes.
Size of float is 4 bytes.

ceil(3.57)=: 4.00
ceil(-3.57)=: -3.00
floor(3.57)=: 3.00
floor(-3.57)=: -4.00
round(3.57)=: 4.00
round(-3.57)=: -4.00
```

Q3 Class Code

```
#include <stdio.h>
#include <string.h>
int main(void)    {
  char text[100];
  double dp=3.57, dn=-3.57;
  int size, i;
  printf("Enter a string: ");
  scanf("%s", text);
  printf("you entered %s\n", text);
  size = strlen( text );
  printf("The number of characters in the input string is %d.\n",size);

text[3]='\0';
  printf("After inserting a null character on the third pos,
          length of string is %d.\n",strlen(text));
  printf("Size of text (char array) is %d.\n",sizeof(text));
  printf("Size of int is %d bytes.\n",sizeof(int));
  printf("Size of char is %d byte.\n",sizeof(char));
  printf("Size of double is %d bytes.\n",sizeof(double));
  printf("Size of float is %d bytes.\n\n",sizeof(float));
  printf("ceil(%0.2f)=: %0.2f\n",dp,ceil(dp));
  printf("ceil(%0.2f)=: %0.2f\n",dn,ceil(dn));
  printf("floor(%0.2f)=: %0.2f\n",dp,floor(dp));
  printf("floor(%0.2f)=: %0.2f\n",dn,floor(dn));
  printf("round(%0.2f)=: %0.2f\n",dp,round(dp));
  printf("round(%0.2f)=: %0.2f\n",dn,round(dn));
  return 0;
 }
```

Output:

```
Enter a string: birdie
you entered birdie
The number of characters in the
input string is 6.
After inserting a null character on
the third pos, length of string is 3.
Size of text (char array) is 100.
Size of int is 4 bytes.
Size of char is 1 byte.
Size of double is 8 bytes.
Size of float is 4 bytes.

ceil(3.57)=: 4.00
ceil(-3.57)=: -3.00
floor(3.57)=: 3.00
floor(-3.57)=: -4.00
round(3.57)=: 4.00
round(-3.57)=: -4.00
```

Q4

- What is the output of the program shown below?

```
#include <stdio.h>
int main(void)
{
        char a1[2] = {'x','y'};
           printf("1. %s \n", a1); // puts(a);
        char a2[3] = {'x', 'y'};
            printf("2. %s \n",a2); //puts(a);
        char a3[2] = "xy";
            printf("3. %s \n",a3); //puts(a);
        char a4[3] = "xy";
            printf("4. %s \n",a4); //puts(a);
        char a5[5] = {'x','y','z','\0','d'};
            printf("5. %s \n",a5); //puts(a);
        char a6[5] = "xyz\0d";
            printf("6. %s \n",a6); //puts(a);
        char a7[5] = "xy\\0";
            printf("7. %s \n",a7); //puts(a);
         return 0;
        }
```

Q4

- What is the output of the program shown below?

```
#include <stdio.h>
int main(void)
{
        char a1[2] = {'x','y'};
           printf("1. %s \n", a1); // puts(a);
        char a2[3] = {'x', 'y'};
            printf("2. %s \n",a2); //puts(a);
        char a3[2] = "xy";
            printf("3. %s \n",a3); //puts(a);
        char a4[3] = "xy";
            printf("4. %s \n",a4); //puts(a);
        char a5[5] = {'x','y','z','\0','d'};
            printf("5. %s \n",a5); //puts(a);
        char a6[5] = "xyz\0d";
            printf("6. %s \n",a6); //puts(a);
        char a7[5] = "xy\\0";
            printf("7. %s \n",a7); //puts(a);
         return 0;
        }
```

```
Output:
1. xy `
2. xy
3. xyxy
4. xy
5. xyz
6. xyz
7. xy\0
```

Q5 Class Exercise

- Write a program that does the following:
  - Declares a character array of size 100.
  - Asks the user to input a string and reads it using scanf() function .
  - Prints the number of characters in the i/p string.
  - Prints the size of the character array.
  - Produces an output as shown below:

```
Enter a string: Magnificent!
you entered "Magnificent!"
The number of characters in the string is 12.
Size of the character array that holds the i/p string is 100.
```

Q5 Class Exercise

- Write a program that does the following:
  - Declares a character array of size 100.
  - Asks the user to input a string and reads it using scanf() function .
  - Prints the number of characters in the i/p string.
  - Prints the size of the character array.
  - Produces an output as shown below:

```
#include <stdio.h>
#include <string.h>
int main(void)
{
 char text[100];
 printf("Enter a string: ");
 scanf("%s", text);
 printf("you entered \"%s\"\n", text);

 printf("The number of characters in the string is %d.\n", strlen(text));
 printf("Size of the character array that holds the i/p string is %d.\n", sizeof(text));
 return 0;
}
```

```
Enter a string: Magnificent!
you entered "Magnificent!"
The number of characters in the string is 12.
Size of the character array that holds the i/p string is 100.
```

Q6 Class Exercise

- Write a program that does the following:
  - Asks the user to input a string and reads it using scanf .
  - Prints the string **using a for loop – one character at a time**.
  - Prints the string in one shot **using the character array address**.
  - Produces an output as shown below:

```
Enter a string: Tenacious!
You entered Tenacious!
Tenacious!
Tenacious!
```

Q6

- Write a program that does the following:
  - Asks the user to input a string and reads it using scanf .
  - Prints the string **using a for loop – one character at a time**.
  - Prints the string in one shot **using the character array address**.
  - Produces an output as shown below:

```
Enter a string: Tenacious!
You entered Tenacious!
Tenacious!
Tenacious!
```

```
#include <stdio.h>
#include <string.h>
int main(void)
{
 char text[100];
 int size, i;
 printf("Enter a string: ");
 scanf("%s", text);
 printf("You entered %s\n", text);
 size = strlen( text );
 for(i = 0; i< size; i++)
            printf("%c", text[i]);

  printf("\n");

printf("%s\n",text);
 }
```

Q7

- Write a program that does the following:
  - Asks the user to input a string and reads it using scanf .
  - Uses a for loop to scan the string – a character at a time and counts upper-case and lower-case letters.
  - Produces an output as shown below:

```
Enter a string: HOppeRR
upper = 4, lower = 3
```

Q7

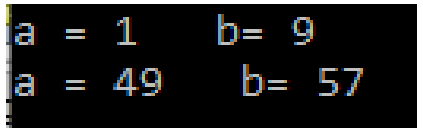
```
#include <stdio.h>
#include <string.h>
int main(void)
{
  char text[100];
  int size, i, upper = 0, lower = 0;
  printf("Enter a string: ");
  scanf("%s", text);
  size = strlen( text );
  for(i = 0; i < size; i++)
  {
    if('A' <= text[i] && text[i] <= 'Z')
                upper++;
    else if('a' <= text[i] && text[i] <= 'z')
                lower++;
  }
      printf("upper = %d, lower = %d\n", upper, lower);
}
```

- Write a program that does the following:
  - Asks the user to input a string and reads it using scanf .
  - Uses a for loop to scan the string – a character at a time and counts upper-case and lower-case letters.
  - Produces an output as shown below:

```
Enter a string: HOppeRR
upper = 4, lower = 3
```

Q8

- Write a program that does the following:
  - Declares variables **a** and **b** of **type char** and assigns them values 1 and 9**.**
  - Prints the contents of a and b.
  - Prints the ascii values of a and b.
  - Produces an output as shown below:



Q8

- Write a program that does the following:
  - Declares variables **a** and **b** of **type char** and assigns them values 1 and 9**.**
  - Prints the contents of a and b.
  - Prints the ascii values of a and b.
  - Produces an output as shown below:

```
a = 1    b= 9
a = 49    b= 57
```

```
#include <stdio.h>
int main(void)
{
char a = '1', b = '9';
    printf("a = %c   b= %c\n",a,b);
    printf("a = %d   b= %d\n",a,b);
    return 0;
}
```

Q9

- Write a program that does the following:
  - Asks for an input string.
  - Finds the length of the string.
  - Checks whether the last element of the string is a number and prints that.
  - Produces an output as shown below:

```
Enter a string: WW3
Found 3 - a number at the end!!
```

```
Enter a string: Hello!
A number was not found at the end of the string!!
```

Q9

- Write a program that does the following:
  - Asks for an input string.
  - Finds the length of the string.
  - Checks whether the last element of the string is a number and prints that.
  - Produces an output as shown below:

```
#include <stdio.h>
#include <string.h>
int main(void)
{
  char text[100];
  int size, i;
  printf("Enter a string: ");
  scanf("%s", text);
  size = strlen( text );
  int x = text[size-1];
  if (x>='0' & x <='9')
     printf("Found %c - a number at the end!!",x );
  else
     printf("A number was not found at the end of the string!!\n\n" );
}
```

```
Enter a string: WW3
Found 3 - a number at the end!!
```

```
Enter a string: Hello!
A number was not found at the end of the string!!
```

Q10

- Write a program that does the following:
  - Asks for an input string.
  - Finds the length of the string.
  - Prints an uppercase version of the string..
  - Produces an output as shown below:

```
Enter a string : Happy Birthday

String in Upper Case = HAPPY BIRTHDAY
```

```
Enter a string : dark chocolate!

String in Upper Case = DARK CHOCOLATE!
```

Q10

- Write a program that does the following:
  - Asks for an input string.
  - Finds the length of the string.
  - Prints an uppercase version of the string..
  - Produces an output as shown below:

```
Enter a string : Happy Birthday

String in Upper Case = HAPPY BIRTHDAY
```

```
Enter a string : dark chocolate!

String in Upper Case = DARK CHOCOLATE!
```

```
#include <stdio.h>
#include <ctype.h>
int main() {
   char s[100];
   int i;
   printf("\nEnter a string : ");
   gets(s);
   for (i = 0; s[i]!='\0'; i++) {
         s[i] = toupper(s[i]);
   }
   printf("\nString in Upper Case = %s", s);
   return 0;
}
```

Q11

- Write a program that does the following:
  - Declares a **character array dst** and assigns it:“It is easy to make a day trip”.
  - Declares a **character array src** and assigns it:“by train”.
  - Calls strcat() function by **strcat(dst, src)**.
  - Produces an output as shown below:

```
Before string concatenation.
dst = It is easy to make a day trip
src = by train.


After string concatenation.
dst = It is easy to make a day trip by train.
src = by train.
```

Q11

- Write a program that does the following:
  - Declares a **character array dst** and assigns it:“It is easy to make a day trip”.
  - Declares a **character array src** and assigns it:“by train”.
  - Calls strcat() function by **strcat(dst, src)**.
  - Produces an output as shown below:

```
Before string concatenation.
dst = It is easy to make a day trip
src = by train.


After string concatenation.
dst = It is easy to make a day trip by train.
src = by train.
```

```
#include <stdio.h>
#include <string.h>
int main() {
    char dst[]="It is easy to make a day trip ";
    char src[]="by train. ";
    printf("\n Before string concatenation.");
    printf("\n dst = %s", dst);
    printf("\n src = %s", src);
    printf("\n\n");
    strcat(dst,src);
    printf("\n After string concatenation.");
    printf("\n dst = %s", dst);
    printf("\n src = %s", src);
    printf("\n\n");
    return 0;
}
```

Q12

- Write a program that does the following:
  - Declares a **character array str** of 20 characters.
  - Declares an **integer variable num** and assigns it a value of 100.
  - Uses **itoa()** function to print 100 in binary, decimal, and hexadecimal notation.
  - Produces an output as shown below:

```
    Binary representation =     1100100
   Decimal representation =         100
Hexadecimal representation =          64
```

Q12

- Write a program that does the following:
  - Declares a ~~**character**~~ **array** **str** of 20 characters.
  - Declares an **integer variable** **num** and assigns it a value of 100.
  - Uses itoa() function to print 100 in binary, decimal, and hexadecimal notation.
  - Produces an output as shown below:

```
     Binary representation =     1100100
    Decimal representation =         100
Hexadecimal representation =          64
```

```
#include <stdio.h>
#include <stdlib.h>
int main(){
   int num=100;
   char str[20];
   // convert to a string with binary representation
   itoa(num,str,2);
   printf("%29s %10s\n","Binary representation = ", str);
   // convert to a string with decimal representation
   itoa(num,str,10);
   printf("%29s %10s\n","Decimal representation = ", str);
   // convert to a string with hexadecimal representation
   itoa(num,str,16);
   printf("%29s %10s\n","Hexadecimal representation = ", str);
   return 0;  }
```

Q12

- Write a program that does the following:
  - Declares a **character arrays str1, str2, and str3** of size 15 each.
  - Assign them the values “100”, “100 apples”, and “100.1 degrees” respectively.
  - Uses atoi() function on str1, str2, and str3, assigns the returned values to num1, num2, and num3 variables respectively and prints the output.
  - Uses atof() function on str3 and prints the output.
  - Produces an output as shown below:

```
atoi(str)
num1 = 100      str1 =100
num2 = 100      str2 =100 apples
num3 = 100      str3 =100.1 degrees

atof(str3)
num3 = 100.1    str3 =100.1 degrees
```

Q13

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

int main()
{
   char str1[15] = "100";
   char str2[15] = "100 apples";
   char str3[15] = "100.1 degrees";
   int num1 = atoi(str1);
   int num2 = atoi(str2);
   int num3 = atoi(str3);
   printf("num1 = %d    str1 =%s \n", num1,str1);
   printf("num2 = %d    str2 =%s \n", num2,str2);
   printf("num3 = %d    str3 =%s \n", num3,str3);
   return 0;
}
```

- Write a program that does the following:
  - Declares a **character arrays str1, str2, and str3** of size 15 each.
  - Assign them the values “100”, “100 apples”, and “100.1 degrees” respectively
  - Uses atoi() function on str1, str2, and str3, assigns the returned values to num1, num2, and num3 variables respectively and prints the output.
  - Uses atof() function on str3 and prints the output.
  - Produces an output as shown below:

```
atoi(str)
num1 = 100    str1 =100
num2 = 100    str2 =100 apples
num3 = 100    str3 =100.1 degrees

atof(str3)
num3 = 100.1   str3 =100.1 degrees
```

Q13

- Print the output of the following program containing escape sequences.

```
#include <stdio.h>
int main() {
    char a[] = "He said \"Hello\"";
    char b[] = "C:\\users\\jane\\note.txt";
    char c[] = "*\n**\n***";
    printf("%s\n",a);
    printf("%s\n",b);
    printf("%s\n",c);
    return 0;
}
```

Q13

- Print the output of the following program containing escape sequences.

```
#include <stdio.h>
int main() {
    char a[] = "He said \"Hello\"";
    char b[] = "C:\\users\\jane\\note.txt";
    char c[] = "*\n**\n***";
    printf("%s\n",a);
    printf("%s\n",b);
    printf("%s\n",c);
    return 0;
}
```

```
He said "Hello"
C:\users\jane\note.txt
*
**
***
```

Q14

- Write a program that does the following:
  - Asks for and accepts two strings.
  - Compares the two strings.
  - Print the output of the program as follows:

```
Enter string 1: Star Fruit
Enter string 2: Star Fruit
The two strings have the same content.
```

```
Enter string 1: Golden Kiwi
Enter string 2: Red Mangoes
The two strings do NOT have the same content.
```

Q14

```
#include <stdio.h>
#include <string.h>
int main(void)
{
 char str1[21] , str2[21];
 printf("Enter string 1: ");
 gets(str1);
 printf("Enter string 2: ");
 gets(str2);
 if   (strcmp(str1,str2)==0){
  printf("The two strings have the same content.\n");
 }
 else {
  printf("The two strings do NOT have the same content.\n");
 }
 return 0;
}
```

- Write a program that does the following:
  - Asks for and accepts two strings.
  - Compares the two strings.
  - Print the output of the program as follows:

```
Enter string 1: Star Fruit
Enter string 2: Star Fruit
The two strings have the same content.
```

```
Enter string 1: Golden Kiwi
Enter string 2: Red Mangoes
The two strings do NOT have the same content.
```