**Labsheet - 2**

1. Type in, compile, and run the following program.

**#include <stdio.h>**

**main( )**

**{**

**float rupees, dollars ;**

**printf(“Conversion program.\n”) ;**

**printf(“Enter dollars: “) ;**

**scanf (“%f ” , &dollars) ;**

**rupees = dollars \* 50 ;**

**printf ( “%f dollars = %f rupees. \n”, dollars, rupees ) ;**

**}**

1. Modify the exchange rate program to read the current exchange rate from the user.
2. Instead of reading the exchange rate from the user, use **#define** to make it a constant.
3. Create a separate file called **exchange**. h and move the **# define** for the exchange rate into this new file. Remove the **# define** from your program file and add the line:

**#include “exchange. h”**

Verify that the program still works correctly.

1. Recally that a \n in the format string of a **printf()** statement causes a newline to be printed. A single format may contain multiple \n’s , so one **printf()** statement does not have to correspond to a one line of output. Write a small program containing a **printf()** statement with several newlines.
2. Sometimes the format string of a **printf ()** statement becomes too long to fit on a single line of the program source file. It is possible to close the format string with a double quote and then open it again on the next line with another double quote. For example:

**#include <stdio .h>**

**main ()**

**{**

**float dollars, pounds ;**

**printf (“\nConversion program. \nThis program reads the”**

**“number of dollars from the user, \ncomputes the”**

**“corresponding amount of rupees , \nand outputs”**

**“ the result . \n\nEnter dollars : “) ;**

**/\* . . . \*/**

**}**

The compiler will automatically concatenate the strings. Try it.

1. A printf () format string can contain serveral other *escape sequences* besides \n. The complete list is:

Sequence Meaning

\a Alert (bell)

\b Backspace

\f Formfeed

\n Newline

\r Carriage return

\t Horizontal tab

\v Vertical tab

\\ Backslash

\? Question mark

\’ Single quote

\” Double quote

Write a program with printf ()statements containing several of these escape sequences.

1. The printf () function returns the number of characters it prints. This return value is most often ignored. Write a program which records the return value of a printf () statement in an integer variable and then prints out the value of the variable. For example:

**int i;**

**i= printf(“12345”) ;**

Add a newline to the string to be printed and run the program again.

1. scanf() returns the number of variables to which it assigns a value. The return value can be used to check whether of not scanf() was successful. For example, if scanf() is trying to read an int but the user types in letters, scanf() will return 0 to indicate it failed. Write a program which reads in an integer and then prints out the integer it reads as well as the return value of scanf(). Run the program twice, once entering a valid integer and once entering alphabetic characters.
2. On a UNIX system, the user can type <control>d to indicate there is no more input (i.e. end-of-file has been reached. If scanf() encounters end -of-file when it starts trying to read input it returns the symbol EOF.

Write a program in which scanf () reads in an int. Assign the return value of scanf() to a variable. When the program is run, enter the appropriate combination of keys to indicate end-of-file on your system. Print out the value returned by scanf(). It should be the same as the same as the preprocessor constant EOF (which is usually -1)

1. Write a program which reads in an amount of seconds from the user and outputs the equivalent amount of minutes and the equivalent number of hours. Use preprocessor constants (i.e #define’s) for the conversion factors. Use integer variables to store the seconds, minutes, and hours. When you run the program, enter 5000 seconds. Notice how the division of one integer by another results in an integer (i.e. the remainder of the division is lost).
2. Modify the time conversion program to use float variables for the number of seconds, minutes and hours. Be sure to modify the printf() and scanf() statements. Print out the results to two decimal places.
3. Write a program which reads in the degrees Fahrenheit from the user and prints out the corresponding degr ees Celsius. The conversion formula is : *C =5\*(F – 32)/9.*
4. Write a program to calculate the area and perimeter of a circle. You should use a Symbolic constant to substitute for the value of pi in the computation.
5. Write a program to read your name from the keyboard and output it on the screen.
6. Write a program to calculate the sum of any two integers and output the result in decimal number system, hexadecimal number system and octal number system.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*