**Lab Sheet 4**

1. Write a program to calculate electricity bill based on the following information.

|  |  |
| --- | --- |
| **Consumption Unit** | **Rate of Charge** |
| 0-150 | Rs. 3 per unit |
| 151 - 350 | Rs. 100 plus Rs.3.75 per unit exceeding 150 units |
| 351 - 450 | Rs. 250 plus Rs.4 per unit exceeding 350 units |
| 451 - 600 | Rs. 300 plus Rs.4.25 per unit exceeding 450 units |
| Above 600 | Rs. 400 plus Rs.5 per unit exceeding 600 units |

1. Write a program to determine whether the year is leap year or not.

[Note: Leap year comes once in every four years E.g. 1996, 2004, 2016

End of century is not always a leap year. E.g. 2100, 2300

Leap year comes after every 400 years. E.g. 2000, 2400, 2800]

1. Write a C program to accept a mark from the user and fix the grade based on the given condition

|  |  |
| --- | --- |
| **Mark(int)** | **Grade(char)** |
| 0 to 50 | F |
| 51 to 70 | D |
| 71 to 80 | C |
| 81 to 90 | B |
| 1. to 100 | A |

1. Write a program to calculate the parking charges of a vehicle. Enter the type of vehicle as a character (like c for car, b for bus, etc) and number of hours as integer. Calculate charges as given below:

* Truck/bus - Rs 20 per hour
* Car - Rs 10 per hour
* Scooter/Motorcycle - Rs 5 per hour

Hint: If your car is parked for 4 hours then parking fee is Rs.40.

1. Write a program to read the month of the year as an integer and display the name of the month.
2. Write a program to find the number of vowels in a given string (Read one character at a time using getchar() example given below) . Print the number of each individual vowels and total number of consonants. Write the program using else-if.

while ((c = getchar()) != EOF) { //This will read on character at time

}// and read till the input is end of file (ctrl+d)

1. Write a program which prints out the character set. Use a **for** statement to loop from 0 to 127. The body of the **for**  loop should be a  **printf()**  statement which prints out the integer controlling the loop. Print this integer twice, once as a decimal ( **%d** format ) and once as a character ( **%c** format ). The output will give you a list of the characters and their corresponding codes. Some of the characters which are not printable but many of them will be recognizable.
2. Write a program which implements a trivial encryption scheme. The program should read a single character using **getchar()**, add the integer value 13 to the character, and output the result. Continue to loop until getchar()returns **EOF**.

Also write a corresponding decryption program to restore the input to its original state.

1. Write a program containing a **for** loop to print your name, the number of times specified by the user. Repeat the exercise using a **while** loop. Try to achieve the same effect using a **do-while** loop ( be careful when the user asks for 0 iterations ).
2. Write a program that inputs a positive integer number *n* and outputs the odd numbers between 1 and *n*. For example, for *n* = 12, program should output:  
   1 3 5 7 9 11
3. Write a C program to accept ten numbers and find the maximum and minimum integers
4. Write a program to find the average of any ‘n’ numbers.
5. Print the multiplication table of any given integer, read from the user.
6. Write a program to convert a decimal number to its binary form.
7. Write a program with a simple *while* loop to accept N integers from the user and find the count of positive numbers, negative numbers and zeros that were entered.
8. Write a program to calculate the sum of squares of first N natural numbers. The number N is read as input from the user. 0 < N < 32767
9. Write a program to calculate sum of cubes of first n natural numbers.
10. Write a program to calculate sum of squares of first n even numbers
11. Write a program to sum the series 1 + + +…
12. Write a program to sum the series 1/12 + 1/22 +…1/n2
13. Write a program to generate the first ‘n’ terms of the following sequence. ‘n’ should be read as input from user.

0, 3, 7, 12, 18........

HINT: 7 = ((3-0) + 1) + 3

1. Write a program to print out all Armstrong numbers between 1 and 500. If sum of cubes of each digit of the number is equal to the number itself, then the number is called an Armstrong number.

For example, 153 = ( 1 \* 1 \* 1 ) + ( 5 \* 5 \* 5 ) + ( 3 \* 3 \* 3 )

1. Write a program to find the factorial value of any number entered through the keyboard.

Note:- 5!=5\*4\*3\*2\*1=120

1. Write a program to print all prime numbers from 1 to 300.
2. Write a menu driven program to display the following menu that offers six options :
3. Read three numbers.
4. Calculate total.
5. Calculate average.
6. Display the smallest.
7. Display the largest value.
8. Exit.

use *switch case* to check the selection and do the task. The program should work in a loop and quit only when the user selects the *Exit* option.

1. Write a program which reads in a character and two integers from the user. The character should be an arithmetic operator **( +, -, \*, / ).** Use a **switch** statement to determine which operator the user entered, perform that operation on the two integers, and output the result. Include a **default** case in the **switch** statement to print out an error message if an invalid operator is entered. Once the program is working, remove one of the **break** statements and run the program to see the effect.
2. Read in the number of times to loop from the user. Write a **for** loop which uses a  **break** statement to exit the loop after the appropriate number of iterations ( leave the test condition empty).
3. The following C Program finds the sum of all digits in the number 1234 :
4. #include<stdio.h>
5. int main()
6. {
7. int d1,d2,d3,d4, n=1234, rev=0, sum=0;
8. d1=n%10; //d1=4(1234%10)
9. rev = rev \* 10 + d1; //rev=4
10. n = n/10; //n=123
11. d2 = n%10; //d2=3(123%10)
12. rev = rev \* 10 + d2; //rev=43((4\*10)+3)
13. n = n /10; //n=12
14. d3 = n % 10; //d3=2(12%10)
15. rev = rev \* 10 + d3; //rev=432((43\*10)+2))
16. n = n /10; //n=1
17. d4 = n % 10; //d4=1(1%10)
18. rev = rev \* 10 + d4; //rev=4321((432\*10)+1))
19. n = n /10; //n=0
20. sum = d1 + d2 + d3 + d4;
21. printf("\nReversed number is %d",rev);
22. printf("\nThe sum of digits is %d",sum);
23. }
    1. Change the value of n from 1234 to 5432 and show the result of each line of execution.
    2. Change the above program to any number from user. Check the results with a sample five digit number.
    3. Using ‘while’ loop, modify the program to find the reverse of any number given.
24. Write a program that reads a positive integer number *m* and then prints the english name of each digit of that number in a single line.  
    For example, for *m* = 147, program should output:  
    one four seven
25. Write a nested for loop that outputs the following 4-by-10 pattern of 0s:

0000000000

0000000000

0000000000

0000000000

1. Write a program to print the following pattern using nested loops

Pass 1:1 2 3 4 5

Pass 2:1 2 3 4 5

Pass 3:1 2 3 4 5

Pass 4:1 2 3 4 5

Pass 5:1 2 3 4 5

1. Write a program to print the following pattern using nested loops

1

1 2

1 2 3

1 2 3 4

1 2 3 4 5

1. Write a program to print the following pattern using nested loops

1

2 2

3 3 3

4 4 4 4

5 5 5 5 5