Acknowledgement

I would like to express our special thanks to my Computer Science teacher, Mrs. Sapna Arora for guidance and the valuable support in making this practical file. I am also grateful to our principal ma'am, Mrs. Hembala for her keen interest, encouragement and the facilities provided to me during the course of this file. During the course of the file I came to learn about so many new things. I would like to thank all those who helped me directly or indirectly towards the completion of this file.

With sincere thanks,
-Shaurya Kharb

Certificate

This is to certify that this practical file is the record of the project work done by SHAURYA KHARB student of class XI-E in the academic year 2022-2023. This file has been carried under the direct supervision of Mrs. Sapna Arora (Subject Teacher).

Flow Of Controls

Q1) Program to accept three integers and print the largest of three.Make use of only if statement.

CODE:

```
x=y=z=0
x=float(input("Enter first number :"))
y=float(input("Enter second number :"))
z=float(input("Enter third number :"))

max=x
if y>max:
    max=y
if z>max:
    max=z
print("Largest Number Is",max)
```

OUTPUT:

```
Enter first number :5.9
Enter second number :7.8
Enter third number :9
Largest Number Is 9.0
```

Q2)Program to test the divisibility of a number with another number (i.e if a number is divisible by another number).

CODE:

```
number1 = int(input("Enter first number :"))
number2 = int(input("Enter second number :"))
remainder = number1 % number2
if remainder == 0:
    print(print1,"is divisible by" ,number2)
else:
    print(number1,"is not divible by",number2)
```

OUTPUT:

```
Enter first number :7089
Enter second number :6
7089 is not divible by 6
```

Q3)Program to find the multiples of a number(the divisor) out of the given five numbers

CODE:

```
print("Enter 5 numbers below")
num1 = float(input("First number :"))
num2 = float(input("Second number :"))
num3 = float(input("Third number :"))
num4 = float(input("Fourth number :"))
num5 = float(input("Fifth number :"))
divisor = float(input("Enter divisor number :"))
count=0
print("Multiples of",divisor,"are:")

remainder = num1%divisor
if remainder == 0:
    print(num1,sep="")
    count+=1

remainder = num2%divisor
if remainder == 0:
    print(num2,sep="")
    count+=1

remainder = num3%divisor
if remainder == 0:
    print(num3,sep="")
    count+=1

remainder = num4%divisor
if remainder == 0:
    print(num4,sep="")
    count+=1

remainder = num5%divisor
if remainder == 0:
    print(num5,sep="")
    count+=1
print()
print(count,"multiples of",divisor,"found")
```

OUTPUT:

```
Enter 5 numbers below
First number :850
Second number :589
Third number :678
Fourth number :400
Fifth number :56
Enter divisor number :3
Multiples of 3.0 are:
678.0

1 multiples of 3.0 found
```

Q4)Program to print the sum of numbers b/w 1 to 7

CODE:

```
sum =0
for n in range(1,8):
    sum+=n
print("Sum of natural numbers <=",n,'is',sum)
```

OUTPUT:

```
Sum of natural numbers <= 7 is 28
```

Q5)Write a program to input a number and print its first and last digit raised to the length of the number(the number of digits in the number is length of the number)

CODE:

```
import math
num = int(input("Enter a number :"))
ln = len(str(num))
lst=num % 10
fst=num//math.pow(10,ln-1)
print("Length of the given number",num,"is",ln)
print("The first digit is ",fst)
print("The last digit is ",lst)
print("First digit raised to the length:",math.pow(fst,ln))
print("Last digit raised to the length:",math.pow(lst,ln))
```

OUTPUT:

```
Enter a number :7632845628
Length of the given number 7632845628 is 10
The first digit is  7.0
The last digit is  8
First digit raised to the length: 282475249.0
Last digit raised to the length: 1073741824.0
```

Q6)Write a python script to print the Fibonacci series' first 20 elements. Some initial elements of the Fibonacci series are : 0112358.....

CODE:

```
first = 0
second = 1
print(first)
print(second)
for a in range(1,19):
    third = first + second
    print(third)
    first,second = second,third
```

OUTPUT:

```
0
1
1
2
3
5
8
13
21
34
55
89
144
233
377
610
987
1597
2584
4181
```

Q7)Write a program to write the lowest and second lowest number from 10 integers

CODE:

```
small=smaller=0
for i in range(10):
    n=int(input("Enter number :"))
    if i == 0:
        small=n
    elif i == 1:
        if n<= small:
         smaller=n

        else:
            smaller=small
            small=n
    else:
        if n<smaller:
            small=smaller
            smaller=n
        elif n<small :
            small=n
    print("The Lowest Number is :",smaller)
    print("The Second Lowest Number is :",small)
```

OUTPUT:

```
Enter number :68
The Lowest Number is : 0
The Second Lowest Number is : 68
Enter number :567
The Lowest Number is : 68
The Second Lowest Number is : 567
```

Q8)Write python script to print the following pattern

```
1
1 3
1 3 5
1 3 5 7
```

CODE:

```
for a in range(3,10,2):
    for b in range(1,a,2):
        print(b,end = '')
    print()
```

OUTPUT:

```
1
13
135
1357
```

Q9)Given a string, say 'Whoa'.Write a program to print as:

```
W
W h
W h o
W h o a
```

CODE:

```
str=''
for x in "Whoa":
    str = str+x
    print(str)
```

OUTPUT:

```
W
Wh
Who
Whoa
```

Q10) Write a program to print the pattern

```
1
2 2
3 3 3
4 4 4 4
5 5 5 5 5
```

CODE:

```
rows=5
for num in range(1,rows+1):
    for i in range (num):
        print(num,end="")
    print()
```

OUTPUT:

```
1
22
333
4444
55555
```

LISTS

Q11) Write a programme that inputs a list, replicates it twice & then prints the sorted list in ascending and descending orders.

CODE:

```
val = eval (input("Enter a list : "))
print("Original List :",val)
val = val*2
print("Replicated list :", val)
val.sort()
print("Sorted in ascending order: ", val)
val.sort(reverse = True)
print("Sorted in descending order: ", val)
```

OUTPUT:

```
Enter a list : [32,11,45]
Original List : [32, 11, 45]
Replicated list : [32, 11, 45, 32, 11, 45]
Sorted in ascending order:  [11, 11, 32, 32, 45, 45]
Sorted in descending order:  [45, 45, 32, 32, 11, 11]
```

Q12) Programme to calculate the mean of a given list of numbers.

CODE:

```
lst = eval(input("Enter List :"))
length = len(lst)
mean = sum = 0
for i in range (0,length) :
    sum += lst[i]
mean = sum/length
print("Given list is :",lst)
print("The mean of the given list is :",mean)
```

OUTPUT:

```
Enter List :[-7,15,46,-85,-4,45,95,0,41,112]
Given list is : [-7, 15, 46, -85, -4, 45, 95, 0, 41, 112]
The mean of the given list is : 25.8
```

Q13) WAP to input 2 lists and display the maximum element from the elements of both of the list combined, along with its index in its list

CODE:

```
lst1 = eval(input("Enter list 1 : "))
lst2 = eval(input("Enter list 2 : "))
mx1 = max(lst1)
mx2 = max(lst2)
if mx1 >= mx2 :
    print(mx1 , "the maximum value is in the list 1 at index", lst1.index(mx1))
else :
    print(mx2 , "the maximum value is in the list 2 at index", lst2.index(mx2))
```

OUTPUT:

```
Enter list 1 : [0,-1,-2,-3,-4,-5]
Enter list 2 : [-10,-9,-8,-7,-6]
0 the maximum value is in the list 1 at index 0
```

Q14) *"L"* is a non-empty list of ints. Print the smallest and largest integer of *L*. Write the code without using a loop.

CODE:

```
L = eval(input("Enter list : "))
length = len(L)
L.sort()
print("smallest : ", L[0])
print("largest : ", L[length-1])
```

OUTPUT:

```
Enter list : [1,2,3,4,5,6,7,8,9,10]
smallest :  1
largest :  10
```

Q15) WAP to input a list of numbers & swap elements at the even places with those at odd places.

CODE:

```
val = eval(input("Enter a list : "))
print("Original list :", val)
s = len(val)
if s%2 != 0 :
    s = s-1
for i in range(0,s,2):
    val[i],val[i+1] = val[i+1], val[i]
print("List after swaping :", val)
```

OUTPUT:

```
Enter a list : [0,1,2,3,4,5,6,7,8,9,10]
Original list : [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
List after swaping : [1, 0, 3, 2, 5, 4, 7, 6, 9, 8, 10]
```

Q16) WAP to input a list and ask for a new element & its position. Then insert the given element at the given position.

CODE:

```
lst = eval(input("Enter a list of integers :"))
print("Original List :", lst)
el = int(input("Enter a new integer element:"))
ln = len(lst)
pos = int(input("Insert at position? <"+str(ln)+":"))
lst.insert(pos,el)
print("List after insertion :", lst)
```

OUTPUT:

```
Enter a list of integers :[0,1,2,3,4,6,7,8,9,10]
Original List : [0, 1, 2, 3, 4, 6, 7, 8, 9, 10]
Enter a new integer element:5
Insert at position? <10:5
List after insertion : [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

Q17) WAP to input a list *lst* & 2 numbers *M* & *N* .Then create a list from those lst elements which are divisible both M & N.

CODE:

```
lst1 = eval(input("Enter a list:"))
lst2 = []
M,N = eval(input("Enter two numbers as M & N : "))
for num in lst1 :
    if (num % M == 0 and num % N == 0):
        lst2.append(num)
print("New created list is :", lst2)
```

OUTPUT:

```
Enter a list:[6,8,11,6,12,7,16]
Enter two numbers as M & N : 6,2
New created list is : [6, 6, 12]
```

Dictionaries

Q18) Write a programme to read roll no. and marks of 4 students and create a dictionary from it having roll no. as keys.

CODE:

```
rno = []
mks = []
for a in range(4):
    r,m = eval(input("Enter roll no. , Marks : "))
    rno.append(r)
    mks.append(m)
d = {rno[0]:mks[0], rno[1]:mks[1], rno[2]:mks[2], rno[3]:mks[3],}
print("Created Dictionary :")
print(d)
```

OUTPUT:

```
Enter roll no. , Marks : 1,50
Enter roll no. , Marks : 2,60
Enter roll no. , Marks : 3,70
Enter roll no. , Marks : 4,80
Created Dictionary :
{1: 50, 2: 60, 3: 70, 4: 80}
```

Q19) Given a dictionary M which stores the marks of the students of class with roll numbers as the keys and marks as the values. Write a program to check if anyone has scored marks as 89.9.

CODE:

```
M = dict({1:46, 2:56,3:44,4:89.9})
if 89.9 in M.values():
    print("Yes someone has scored 89.9 marks")
else:
    print("No one has scored 89.9")
```

OUTPUT:

```
Yes someone has scored 89.9 marks
```

Q20) WAP to create a dictionary containing names of competition winner students as keys and number of their wins as values.

CODE:

```
n = int(input("How many students? : "))
compWinners = {}
for a in range(n):
    key = input("Name of the student : ")
    value = int(input("Number of Competetinos won : "))
    compWinners[key] = value
print("The dictionary now is : ")
print(compWinners)
```

OUTPUT:

```
How many students? : 3
Name of the student : A
Number of Competetinos won : 1
Name of the student : B
Number of Competetinos won : 2
Name of the student : C
Number of Competetinos won : 3
The dictionary now is :
{'A': 1, 'B': 2, 'C': 3}
```

Q21) Consider the already created dictionary M that stores roll numbers and marks. Write a program to input a roll number and delete it from the dictionary. Display error message if the roll number does not exist in the dictionary.

CODE:

```
M = dict({1:46, 2:56,3:44,4:80,5:87})
print("Created Dictionary :")
print(M)
rno = int(input("Roll no. to be deleted? : "))

if rno in M :
    del M[rno]
    print("Roll no.", rno, "deleted from dictionary.")
else :
    print("Roll no.", rno, "does not exist in dictionary")
print(" Final Dictionary : ")
print(M)
```

OUTPUT:

```
Created Dictionary :
{1: 46, 2: 56, 3: 44, 4: 80, 5: 87}
Roll no. to be deleted? : 3
Roll no. 3 deleted from dictionary.
 Final Dictionary :
{1: 46, 2: 56, 4: 80, 5: 87}
```

Q22) Write a program to create a dictionary with the roll number, name and marks of n students in a class and display the names of students who have marks above 75.

CODE:

```
n = int(input("How many Students?"))
stu={}

for i in range(1, n+1):

    print("Enter details of Student", (i))
    rollno= int(input("Roll number :"))
    name = input("Name")
    marks = float(input("Marks :"))
    d = {"Roll_no" :rollno, "Name": name, "Marks": marks}

    key = "Stu" + str(i)
    stu[key]=d
print("Students with marks> 75 are:")

for i in range(1, n+1):
    key = "Stu" + str(i)

    if stu[key]["Marks"]> 75 :
        print(stu[key])
```

OUTPUT:

```
How many Students?4
Enter details of Student 1
Roll number :1
NameA
Marks :90
Enter details of Student 2
Roll number :2
NameB
Marks :80
Enter details of Student 3
Roll number :3
NameC
Marks :70
Enter details of Student 4
Roll number :4
NameD
Marks :60
Students with marks> 75 are:
{'Roll_no': 1, 'Name': 'A', 'Marks': 90.0}
{'Roll_no': 2, 'Name': 'B', 'Marks': 80.0}
```

Q23) WAP to find common keys : value pairs from 2 dictionaries and shared keys with different items.

CODE:

```
d1 = {"A":1,"B":2,"C":3,"D":4,"E":5}
d2 = {"A":1,"B":4,"C":33,"E":5}
com = {}
keylist = []
for key in d1 :
    item1 = d1[key]
    if key in d2 :
        item2 = d2[key]
        if item1 == item2:
            com[key] = item1
        else:
            keylist.append(key)
print("Two dictionries are : ")
print(d1)
print(d2)
print("------")
print("Common key : value pairs")
print(com)
print("Shared keys with different items :", keylist)
```

OUTPUT:

```
Two dictionries are :
{'A': 1, 'B': 2, 'C': 3, 'D': 4, 'E': 5}
{'A': 1, 'B': 4, 'C': 33, 'E': 5}
------
Common key : value pairs
{'A': 1, 'E': 5}
Shared keys with different items : ['B', 'C']
```

Q24) WAP to check if a dictionary is empty or not.

CODE:

```
dct = eval(input("Enter Dictionary : "))
if dct == dict():
    print("You have entered an empty dictionary")
```

OUTPUT:

```
Enter Dictionary : {}
You have entered an empty dictionary
```

Q25) WAP to create a third dictionary from two dictionaries having some common keys, in such a way so that the values of common keys are added in the third dictionary.

CODE:

```
dct1 = {'1': 100, 2: 200, 3:300}
dct2 = {'1': 300, 2: 208, 5:400}
dct3 = dict(dct1)

dct3.update(dct2)

for i, j in dct1.items():
    for x, y in dct2.items():
        if i==x:
            dct3[i] = (j+y)
    print("Two given dictionaries are:")
    print (dct1)
    print(dct2)
    print("The resultant dictionary")
    print(dct3)
```

OUTPUT:

```
Two given dictionaries are:
{'1': 100, 2: 200, 3: 300}
{'1': 300, 2: 208, 5: 400}
The resultant dictionary
{'1': 400, 2: 408, 3: 300, 5: 400}
```

Q26) WAPP to find the highest 2 values in a dictionary.

CODE:

```
numbers = {31 : 111, 22 : 222, 43 : 333, 14 : 444, 25 : 555}

s = sorted(numbers.values())
print("Given dictionary is : ", numbers)
print("highest two values of a given dictionary are : ", s[-1], s[-2])
```

OUTPUT:

```
Given dictionary is :  {31: 111, 22: 222, 43: 333, 14: 444, 25: 555}
highest two values of a given dictionary are :  555 444
```

Database Concepts

A database system is basically a computer based record keeping system. The collection of data, usually referred to as the database, contains information about one particular enterprise.

A database may also be defined as a collection of interrelated data stored together to serve multiple applications.

Limitations of a File Based Approach

(i) Difficulty in Access

(ii) Data Redundancy : refers to storage of same data multiple times (i.e duplicate data)

(iii) Data Inconsistency : multiple mismatched copies of the same data

(iv) Data Isolation : refers to a situation where data of one file cannot be mapped to other related file in the absence of links or mapping or common formats.

(v) Data Dependence : the close relationship between data stored in files and the software programs that update and maintain those files is called data dependence

(vi) Data Sharing Security/Control Issues

Advantages of a Database

1) Databases reduce the data redundancy to a large extent.

2) Databases can control data inconsistency to a large extent.

3) Databases facilitate sharing of data. Also databases can control which user can view the data.

4) Databases enforce standardised structure & format.

5) Databases can ensure data security.

6) Databases ensure data independence, i.e. any changes to the data structures are handled by the DBMS and application programs are unaffected from it.

DBMS Key Concepts

1) **Database Schema :** A database schema is a sketch/skeleton/blueprint of a planned data. It represents the design of tables, columns, relations, constraints and relationships that make up the logically distinct section of a database
2) **Database Instance :** A database instance is a snapshot of a database that exists as a particular time.

3) **Metadata :** Meta-data refers to data about data, it is stored in a data dictionary. A data dictionary is a file storing metadata of the objects of a database.
4) **Database Constraints :** It is a set of rules that define valid data.
5) **Data Manipulation :** Data manipulation takes place when the data in tables is inserted, updated or deleted.

6) **Database Engine :** A database engine is the underlying software component that a DBMS uses to create , read, update and delete data from a database.

The ***rows*** and ***columns*** of a relation are known as ***Tupples*** & ***Attributes*** respectively.
The number of attributes in a relation is called ***Degree*** and the number of brown in a relation is known as ***Cardinality***

Keys in a Database

1) **Primary Key :** A primary key is a set of more attributes that can uniquely identify tuples within the relation

2) **Candidate Key :** All attribute combinations inside a relation that can serve as a primary key are Candidate key.

3) **Alternate Key :** A candidate key that is not the primary key is called an alternate key

4) **Foreign Key :** A non-key attribute whose values are derived from the primary key of some other table is known as foreign key in its current table

SQL

Q27)For the class representative post, 3 students are the candidates. Your teacher has code named as a band c.Each student of the class has to vote to select the class representative from them. The voting data is available in the form of the following dictionary, which stores which roll number voted for whom.

votes = {1: 'a', 2:'a', 3: 'c', 4: 'a', 5: 'c', 6: 'b', 7: 'b', 8: 'b',\
9:'a',10:'c', 11: 'a', 12: 'b', 13: 'b', 14: 'c', 15: 'b'}

Write a program to calculate total votes for each of the candidates and declare the winner.

CODE:

```
votes = {1: 'a', 2:'a', 3: 'c', 4:'a', 5: 'c', 6: 'b', 7: 'b', 8: 'b', 9: 'a', 10: "c",\
11:"a", 12: 'b', 13: 'b', 14: 'c', 15: 'b'}

result = {}

for k in votes:
    val=votes[k]

    if val not in result:
        result[val] = 1
    else:
        result[val] += 1
print("Voting result is: ")
print(result)
mx=0
for k in result:
    if result[k] >mx:
        mx=result[k]
        winner = k
print("The winner is", winner, "with", mx, "votes.")
```

OUTPUT:

```
Voting result is:
{'a': 5, 'c': 4, 'b': 6}
The winner is b with 6 votes.
```

Q28) Create another table JOB with the attributes as Job_id, Job_des, Emp_id, where our Job_id is the primary key and Job_des, Emp_id cannot left blank and Emp_id is our foreign key here that is related to ID column of earlier created table Employee.

CODE:

```
mysql> CREATE TABLE Job
    -> (Job_id integer NOT NULL PRIMARY KEY,
    -> Job_des Varchar(30) NOT NULL,
    -> Emp_id integer REFERENCES Employee(ID));
```

OUTPUT:

```
Query OK, 0 rows affected (0.14 sec)
```

Q29) Write SQL command to create a table, new table "WORK", which has exactly the same structure as the "JOB" table created in the previous question (q4). Show the table structure of the newly created table.

CODE:

```
Database changed
mysql> CREATE TABLE WORK
    -> (SELECT* FROM JOB);
Query OK, 0 rows affected (0.34 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> DESC WORK;
```

OUTPUT:

```
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| Job_id  | int         | NO   |     | NULL    |       |
| Job_des | varchar(30) | NO   |     | NULL    |       |
| Emp_id  | int         | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

Q30) Write SQL command to create a table, new table "STAFF" with only ID, First_Name, Last_Name fields from the employee table of question 3, Also show the structure of the newly created table.

CODE:

```
mysql> create table staff
    -> (SELECT ID,First_Name,Last_Name
    -> from Employee);
Query OK, 0 rows affected (0.18 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

OUTPUT:

```
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| ID         | int         | NO   |     | NULL    |       |
| First_Name | varchar(30) | NO   |     | NULL    |       |
| Last_Name  | varchar(30) | NO   |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

Q31) Perform a simple calculation using SQL to find the output of

1+6 and then perform 3 x 8 using the dummy table "DUAL" available in sql.

CODE:

```
mysql> SELECT 1+6;
+-----+
| 1+6 |
+-----+
|   7 |
+-----+
```

OUTPUT:

```
mysql> SELECT 3*8 FROM DUAL;
+-----+
| 3*8 |
+-----+
|  24 |
+-----+
1 row in set (0.00 sec)
```