INDEX

Name: Shaurya Rathi Class: XI Year: 2023-24

Expt. No.	Date	Title of the Experiments	Page No.	Date of Submission	Remarks
1.	17-1-24	Computing x^n	1		
2.	17-1-24	Simple Interest	2		
3.	17-1-24	Even or Odd	3		
4.	17-1-24	Grading System	4		
5.	17-1-24	Fibonacii Numbers	5		
6.	17-1-24	Prime Numbers	6		
7.	24-1-24	Armstrong Numbers	7		
8.	24-1-24	Perfect Number	8		
9.	14-2-24	Sum of Series	9		
10.	14-2-24	Pattern	10		
11.	14-2-24	Palindrome	11		
12.	14-2-24	Character Count	12.		
13.	14-2-24	Capitalising	13		
14.	14-2-24.	Second Largest Element	14		
15.	24-2-24	Selection of Elements	15.		
16.	24-2-24	Cumulative Sum	16		
17.	24-2-24	Sorting	17		
18.	28-2-24	Max & Min	18		
19.	28-2-24	Frequency of Element	19		
20.	1-3-24	Square of Tuple	20.		
21.	1-3-24	Extraction of Tuple	21.		
22.	1-3-24	Linear Search	22.		
23	1-3-24	Bubble Sort	23		

INDEX

Name..... Class..... Year.....

Expt. No.	Date	Title of the Experiments	Page No.	Date of Submission	Remarks
24	1-3-24	Values in Dictionary	24		
25	1-3-24	Customer Database	25		
26	2-3-24	SQL Problems	27		

OUTPUT:

```
Enter base number : 5
Enter exponent : 2
Answer = 25
```

Name of the Experiment : Exponent Date : 17/01/2024 Page No. : 1

Experiment No. : 1 Experiment Result :

"""Write a program to compute x^n, given x, n.

Code : '''

```
x = int(input("Enter base number:"))
n = int(input("Enter exponent:"))

print("Answer = " + str(x**n))
```


Teachers sign .....

OUTPUT:

```
Enter principal amount: 100
Enter time (in years): 1
Enter rate per annum: 5
Simple Interest = 5
Final amount = 105
```

BS
17/1/24

Name of the Experiment : Simple Interest Date : 17/01/2024 Page No. : 2

Experiment No. : 2 Experiment Result :

"""Write a program to compute Simple Interest

CODE: """

```
p = float(input("Enter principal amount: "))
t = float(input("Enter time (in years): "))
r = float(input("Enter rate per annum: "))

s = p * t * r / 100
amt = p + s

print("Simple Interest = ", s)
print("Final amount =", amt)
```

OUTPUT 1 –

Enter a number: 5
Number is odd.

OUTPUT 2 –

Enter a number: 0
Number is even.

17/1/25

Name of the Experiment : Even or odd Date : 17/01/2024 Page No. : 3

Experiment No. : 3 Experiment Result :

"""Write a program to check whether a number is even or odd.

CODE:"""

```
num = int(input("Enter a number:"))

if num % 2 == 0:
    print("Number is even.")
else:
    print("Number is odd.")
```


Teachers sign .....

OUTPUT ①-

Enter percentage: 72
Grade: B

OUTPUT ②-

Enter percentage: 90
Grade: A

17/12/24

Name of the Experiment : Grading System Date : 17/01/2024 Page No. : 4

Experiment No. : 4 Experiment Result :

"""Write a program to accept percentage and display grade accordingly"""

CODE: """

```
p = float(input("Enter percentage: "))

if p >= 85:
    print("Grade: A")
elif p > 70:
    print("Grade: B")
elif p > 60:
    print("Grade: C")
elif p > 45:
    print("Grade: D")
else:
    print("Grade: F")
```

OUTPUT:

```
Enter limit : 7
0
1
1
2
3
5
```

Name of the Experiment : Fibonacci Series Date : 17/01/2024 Page No. : 5

Experiment No. : 5 Experiment Result :

"' Write a program to display fibonacci series upto a given number.

CODE: "'

```
n = int(input("Enter limit : "))
a = 0
b = 1
c = a + b
if c <= n:
    print(a)
    print(b)
    while c <= n:
        print(c)
        a = b
        b = c
        c = a + b
elif b <= n:
    print(a)
    print(b)
elif a <= n:
    print(a)
else:
    print("Error")
```


Teachers sign .....

OUTPUT ①:

Enter number: 7
2
3
5
7

OUTPUT ②:

Enter number: 0

OUTPUT ③:

Enter number: 2
2

24/11/24

Name of the Experiment : Prime Numbers Date : 17/01/2024 Page No. : 6

Experiment No. : 6 Experiment Result :

" Write a program to ~~check if a number is prime~~ to display primes upto a number.

CODE: '''

```
num = int(input("Enter number:"))
i = 2, b = 0
while i <= num:
    for j in range(2, i-1):
        if i % j == 0:
            b += 1
    if b == 0:
        print(i)
    b = 0
    i += 1
```


Teachers sign .....

OUTPUT ①:

Enter a number: 153
Armstrong number.

OUTPUT ②:

Enter a number: 26
Not an Armstrong number

26/11/24

Name of the Experiment : Armstrong Numbers Date : 24/12/2024 Page No. : 7

Experiment No. : 7 Experiment Result :

```
num = int(input
''' Write a program to check if a number
    is an armstrong number

CODE: '''

num = int(input("Enter a number:"))
n = num
dig = 0
sum = 0

while n > 0:
    n = int(n/10)
    dig += 1
n = num
while n > 0:
    sum += (n % 10) ** dig
    n = int(n/10)

if sum == num:
    print("Armstrong number.")
else:
    print("Not an Armstrong number.")
```


Teachers sign .....

OUTPUT ①:

Enter a number: 28
Perfect number.

OUTPUT ②:

Enter a number: 12
Not a perfect number.

SS
24/1/24

Name of the Experiment : Perfect Numbers Date : 24/01/2024 Page No. : 8

Experiment No. : 8 Experiment Result :

''' Write a program to check whether a number is a perfect number.

CODE: '''

```
num = int(input("Enter a number: "))
sum = 0

for i in range(1, num + 1):
    if num % i == 0:
        sum += i

if sum == 2 * num:
    print("Perfect number.")
else:
    print("Not a perfect number.")
```

OUTPUT ①:

Enter base number : 4
Enter number of terms : 2
Sum of series = 5.0

OUTPUT ②:

Enter base number : 5
Enter number of terms : 3
Sum of series = 18.5

14/2/24

Name of the Experiment : Sum of Series Date : 14/02/2024 Page No. : 9

Experiment No. : 9 Experiment Result :

''' Write a program to print the sum of series:

$$1 + \frac{x}{1!} + \frac{x^2}{2!} + \dots + \frac{x^n}{n!}$$, given x & n.

CODE: '''

```
def fact(n):
    fact = 1
    for i in range(1, n+1):
        fact *= i
    return fact

x = int(input("Enter base number: "))
n = int(input("Enter number of terms: "))
sum = 0

for i in range(n):
    sum += (x**i) / fact(i)

print("Sum of series = ", sum)
```


Teachers sign .....

OUTPUT:

```
1
1 2
1 2 3
```

14/2/24

Name of the Experiment : Pattern Date : 14022024 Page No. : 10

Experiment No. : 10 Experiment Result :

"'Write a program to generate the following pattern :

```
1
1 2
1 2 3
```

CODE: '''

```
for i in range(3):
    for j in range(i+1):
        print(j+1, end = ' ')
    print()
```

OUTPUT ①:

Enter a string: racecar
String is a palindrome.

OUTPUT ②:

Enter a string: helmet
String is not a palindrome.

SS
14/2/24

Name of the Experiment : Palindrome Date : 14/02/2024 Page No. : 11

Experiment No. : 11 Experiment Result :

"" W.A.P. to accept a string and check if it is a palindrome.

CODE: ""

```
str1 = input("enter a string: ")
str2 = ""

for i in range(len(str1)-1, -1, -1):
    str2 += str1[i]

if str2 == str1:
    print("String is a palindrome.")
else:
    print("String is not a palindrome.")
```


Teachers sign .....

OUTPUT:

Enter string: Hello64#

There are 4 lower case, 1 upper case, 2 digits and 1 special characters.

SS
14/2/24

Name of the Experiment : Character Count Date : 14 02 2024 Page No. : 12

Experiment No. : 12 Experiment Result :

" WAP to count lower/upper case, digit, and special characters in a string.

CODE: "'

```
str = input("Enter string: ")
l, u, d, s = 0, 0, 0, 0

for ch in str:
    if ch.islower():
        l += 1
    elif ch.isupper():
        u += 1
    elif ch.isdigit():
        d += 1
    else:
        s += 1
print()
print("There are", l, "lower cases,", u, "upper cases",
      d, "digits and", s, "special characters.")
```


Teachers sign .....

OUTPUT:

Enter a sentence: test sentence
Original string: test sentence
Final string: test Sentence

SS
14/2/24

Name of the Experiment : Capitalising Date : 14022024 Page No. : 13

Experiment No. : 13 Experiment Result :

"" WAP to capitalise the first letter of each word in a string

CODE: '''

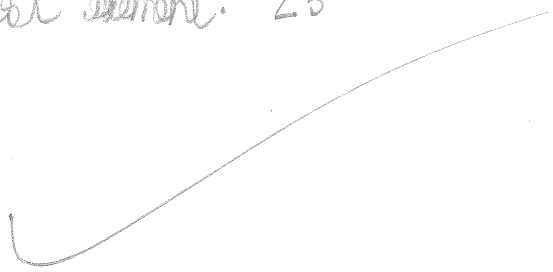
```
str = input("enter a sentence: ")
print("Original string: ", str)
x = str.split()
str1 = ""

for a in x:
    str1 += a.capitalize() + " "

print("Final string: ", str1)
```

Bafna Gold

Teachers sign .....

OUTPUT:

Second largest element: 23



SS
14/2/24

Name of the Experiment : Second Largest Element Date : 14/02/2024 Page No. : 14

Experiment No. : 14 Experiment Result :

''' WAP to display second largest element in a given list [41, 6, 9, 13, 4, 23]

CODE: '''

```
l = [41, 6, 9, 13, 4, 23]

m = max(l)
i = l.index(m)
l.pop(i)

print("Second largest element: ", max(l))
```

Teachers sign .....

OUTPUT:

```
Enter no. of elements : 4
Enter element 1 : 3
Enter element 2 : 19
Enter element 3 : 22
Final list : [22]
```

SS
24/2/24

Name of the Experiment : Selection of Elements Date : 24/02/2024 Page No. : 15

Experiment No. : 15 Experiment Result :

"" WAP to remove all odd numbers from a list

CODE: ""

```
l1 = list()
n = int(input("Enter no. of elements: "))
l = 0
for i in range(n):
    l = int(input("Enter element", i+1, ": "))
    l1.append(l)
l2 = list()
for a in l1:
    if a%2 == 0:
        l2.append(a)

print("Final list:", l2)
```

Teachers sign .....

OUTPUT:

```
Enter no. of elements :  5
Enter element 1 :  7
Enter element 2 :  12
Enter element 3 :  3
Enter element 4 :  25
Enter element 5 :  10
[7, 12, 3, 25, 10]
[7, 19, 22, 47, 57]
```

Good

88

24 | 2 | 24

Name of the Experiment : Cumulative Sum Date : 24/02/2024 Page No. : 16

Experiment No. : 16 Experiment Result : .....

"" WAP to find cumulative sum.

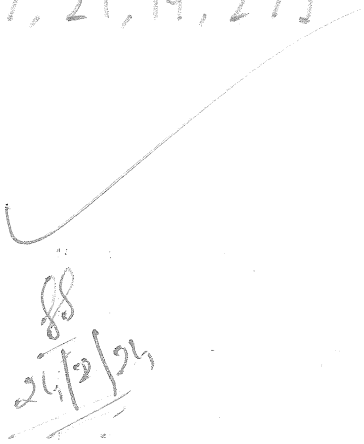
CODE: ""

```
l1, l2 = [], []
n = int(input("Enter no. of elements: "))
s = 0
k = 0
for i in range(n):
    k = int(input("Enter element", i+1, ":"))
    s += k
    l1.append(k)
    l2.append(s)

print(l1)
print(l2)
```

Teachers sign .....

OUTPUT:

```
Enter list: 7, 10, 19, 21, 27
Original list: [7, 10, 19, 21, 27]
Final list: [10, 7, 21, 19, 27]
```



Name of the Experiment : Sorting Date : 24022024 Page No. : 17

Experiment No. : 17 Experiment Result :

''' WAP to swap elements at even and odd positions.

CODE : '''

```
l = list(input("Enter list: "))
print("Original list:", l)

length = len(l)
if length % 2 != 0:
    length -= 1

for i in range(0, length, 2):
    l[i], l[i+1] = l[i+1], l[i]

print("Final list: ", l)
```

Teachers sign .....

OUTPUT:

Enter list : 10, 3, 29, 108, 72, 1
Largest element: 108
Smallest element: 1

28/2/24

Name of the Experiment : Max and Min Date : 28/02/2024 Page No. : 18

Experiment No. : 18 Experiment Result :

''' WAP to find largest and smallest element in a list.

CODE: "

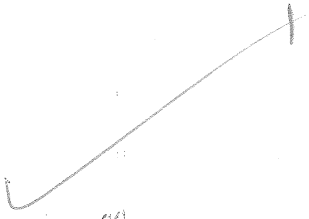
```
l1 = list(input("Enter list:"))
n = len(l1)
min, max = l1[0], l1[0]

for i in range(n):
    if min > l1[i]:
        min = l1[i]
    if max < l1[i]:
        max = l1[i]

print("Largest element:", max)
print("Smallest element:", min)
```

Teachers sign .....

OUTPUT:

Enter list: 5, 7, 12, 12, 5, 25

Element	Frequency
5	2
7	1
12	2
25	1



28/2/24

Name of the Experiment : Frequency of element Date : 29/02/2024 Page No. : 19

Experiment No. : 19 Experiment Result :

"" WAP to display frequencies of all the elements in the list

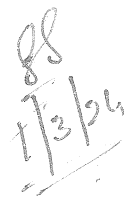
CODE: ""

```
l = list(input("enter list :"))
l1, l2 = [], []
x = 0
for i in l:
    if i not in l1:
        x = l.count(i)
        l1.append(i)
        l2.append(x)

print("Element \t\t\t Frequency")
for i in range(len(l1)):
    print(l1[i], "\t\t\t", l2[i])
```

Teachers sign .....

OUTPUT:

Enter tuple : 1, 2, 3, 4, 5
(1, 4, 9, 16, 25)



Name of the Experiment : Square of Tuple Date : 01032024 Page No. : 20

Experiment No. : 20 Experiment Result :

""" WAP to accept a tuple and square each element

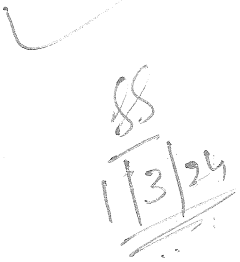
CODE: """

```
t = eval(input("Enter tuple: "))
l = list(t)
for i in range(len(l)):
    l[i] = l[i] **2
t_new = tuple(l)
print(t_new)
```

Teachers sign .....

OUTPUT:

Enter string elements: ('abc', '27', 'name', 21)
[27, 21]



Name of the Experiment : Extraction of tuple Date : 01032024 Page No. : 21

Experiment No. : 21 Experiment Result :

''' WAP to accept a tuple of strings and display the numerical values in it.

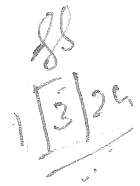
CODE: '''

```
t = eval(input("Enter string elements: "))
l = []

for i in t:                    # extracting digits
    if i.isdigit():
        l.append(i)

for i in range(len(l)):        # Converting into integers
    l[i] = int(l[i])

print(l)
```

Teachers sign .....

OUTPUT:

Enter list : [25, 91, 28, 900, 85]
Enter item to be searched : 28
Element found at position 3.



Name of the Experiment : Linear Search Date : 03/03/2024 Page No. : 22

Experiment No. : 22 Experiment Result :

""" WAP to perform linear search

CODE : "

```
l = eval(input("enter list : "))
item = int(input("Enter item to be searched : "))
loc = -1

for i in range(len(l)):
    if l[i] == item:
        loc = i+1
        break

if loc > -1:
    print("Element found at position", loc)
else:
    print("element not found.")
```

Teachers sign .....

OUTPUT:

Enter list to be sorted: [29, 31, 28, 900, 22]
Bubble sorted list: [22, 28, 29, 31, 900]

Name of the Experiment : Bubble Sort Date : 01032024 Page No. : 23

Experiment No. : 23 Experiment Result :

"" WAP to perform bubble sort

CODE: ""

```
l = eval(input("Enter list to be sorted: "))

for i in range(len(l)-1):
    for j in range(len(l)-1):
        if l[j] > l[j+1]:
            l[j], l[j+1] = l[j+1], l[j]

print("Bubble sorted list:", l)
```

Teachers sign .....

OUTPUT:

```
Enter number of students: 3
Enter details of student 1
Name: Viraj
Percentage: 90
Enter details of student 2
Name: Stavya
Percentage: 85
Enter details of student 3
Name: Shubhan
Percentage: 95
Highest values: 95,90
```

Good

88

1F324

Name of the Experiment : Values in Dictionary Date : 01032024 Page No. : 24

Experiment No. : 24 Experiment Result :

''' WAP to print the 2 largest values in a dictionary

CODE: '''

```
n = int(input("Enter number of students: "))
sd = {}
s_name = ""
per = 0

for i in range(n):
    print("Enter details of student", i+1)
    s_name = input("Name:"))
    per = int(input("Percentage:")
    sd[s_name] = per

s = sorted(sd.values())
print("Highest values:", s[-1], s[-2])
```


Teachers sign .....

OUTPUT:

```
How many customers : 3
Enter details of customer 1:
Enter name : Akshith
Enter phone number : 9922 8845
Enter details of customer 2:
Enter name : Viraj
Enter phone number : 8877 6262
Enter details of customer 3:
Enter name : Sanvi
Enter phone number : 7766 7788
1. Show Record
2. Add new record
3. Delete a customer
4. Search Record
5. Sort Record
6. Update record
7. Exit
Enter choice : 3
Enter name to be deleted : Akshith
Akshith is deleted successfully.
```

Name of the Experiment : Customer Database Date : 01/03/2024 Page No. : 25

Experiment No. : 25 Experiment Result :

" Write a menu driven program to perform the following operations -
Show record, Add new customer, Delete a customer, Search record, Update record, Sort record, Exit

CODE: "

```
n = int(input("How many customers: "))
cust = {}
for i in range(n):
    print("Enter details of customer", i+1, ":")
    c_name = input("Enter name of customer: ")
    ph_no = int(input("Enter phone number: "))
    cust[c_name] = ph_no
c = 0

while c != 7:
    print(""" 1. Show record
              2. Add new record
              3. Delete a customer
              4. Search record
              5. Sort record
              6. Update record
              7. Exit """)

    c = int(input("Enter choice: "))
```

Teachers sign .....

```
if c == 1:
    for i in cust:
        print(i, ":", cust[i])
elif c == 2:
    print("Enter details of new customer:")
    c_name = input("Enter name:")
    ph_no = int(input("Enter phone number:"))
    cust[c_name] = ph_no
elif c == 3:
    name = input("Enter name to be deleted:")
    f = cust.pop(name, -1)
    if f != -1:
        print(f, "is deleted successfully.")
    else:
        print(f, "not found.")
elif c == 4:
    name = input("Enter customer name:")
    if name in cust:
        print(name, "found in the record.")
    else:
        print(name, "not found in the record.")
elif c == 5:
    l = sorted(cust)
    for i in l:
        print(i, ":", l[i])
elif c == 6:
    c_name = input("Enter name:")
    ph = int(input("No. to be modified."))
```

Teachers sign .....


```
elif c == 7:
    break
```

Name of the Experiment : SQL Problems Date : 02032024 Page No. : 27

Experiment No. : 26 Experiment Result :

SQL Problems

I Generate an electricity bill -

1. Create a table with each of the following fields:

Field name	Type
RR number	varchar2(10)
Consumer name	varchar2(20)
Bill date	date
Units	int

2. Insert 10 records into table
3. Check its structure.
4. Add the following fields to the table.

Field name	Type
Bill amount	Float
Due date	date

5. Compute the bill amount - Min amt. - ₹ 50

First 100 units	₹ 4.5
For more than 100	₹ 5.5

6. Compute due date as bill date + 15 days
7. List all the bills generated

Bafna Gold

Teachers sign .....

Field	Type	Null	Key	Default	Extra
rrn	varchar(10)	YES	-	NULL	-
cname	varchar(20)	YES	-	NULL	-
billdate	date	YES	-	NULL	-
units	int	YES	-	NULL	-

Name of the Experiment :..... Date : Page No. : 28

Experiment No. :..... Experiment Result :.....

Solution -

1. Creating table :

```
create table e_bill ( srn varchar2(10),
                      cname varchar2(20),
                      billdate date,
                      units int );
```

Table Created.

2. Inserting records -

```
insert into ebill values ('a1', 'Krishna', '2-mar-2024', 31);
insert into e-bill values ('a2', 'Vittal', '2-mar-2024', 50);
insert into e-bill values ('a3', 'Pandu', '2-mar-2024', 75);
```

3. Checking structure of table -

```
desc e-bill;
```

4. Inserting billds -

```
alter table e_bill add (billamt float, duedate date);
```

5. Calculating bill amount -

```
update e-bill set billamt = 50;
update e-bill set billamt = 50 + 100*4.5 + (units-100)*
                            5.5 where units > 100;
update e_bill set billamt = 50 + units*4.5 where units <=10
```

Bafna Gold

Teachers sign .....

s.no.	cname	billdate	units	billamt	duedate
s1	Krishna	2024-03-02	31	187.5	2024-03-17
s2	Vittal	2024-03-02	50	275	2024-03-17
s3	Pandu	2024-03-02	75	387.5	2024-03-17

6. Setting ~~due~~ due date -

```
update e-bill set duedate = billdate + 15;
```

7. Displaying the table -

```
select * from e-bill;
```

II Student Database -

1. Create a table for a class of students -

Field name	Data type
Student ID	int
Student name	varchar(50)
Economics	int
Computer	int
English	int
Sanskrit	int

2. Add records into the table.
3. Display description of the fields.
4. Alter the table to calculate total & percentage.
5. Compute result as 'pass' or 'fail' if marks ≥ 35 for all sub
6. List contents of table
7. List only student id and students name
8. List students who have passed or failed.
9. Count no. of students passed or failed.

Table	Column	Data type	Length	Decimal	Scale	Primary key	Nullable	Unique	Composite
s-db	s_id	Number	-	-	0	-	✓	-	-
	s_name	varchar	20	-	-	-	✓	-	-
	Economics	Number	-	-	0	-	✓	-	-
	Computer	Number	-	-	0	-	✓	-	-
	English	Number	-	-	0	-	✓	-	-
	Sanskrit	Number	-	-	0	-	✓	-	-

10. List students with percentage greater than 60
11. Sort the table according to student id.

Solution -

1. Creating table :

```
create table s_db ( s_id int, s_name varchar(50),
                    Economics int, Computer int,
                    English int, Sanskrit int);
```

2. Inserting records :

```
insert into s_db values (01, 'Krishna', 67, 35, 20, 90);
insert into s_db values (2, 'Ram', 80, 40, 90, 70);
insert into s_db values (3, 'Vittal', 67, 24, 45, 59);
```

3. Describing the table :

```
desc s_db;
```

4. Altering table :

```
alter table s_db add (total int, perc float, result varchar
update s_db set total = Economics + Computer
                        + English + Sanskrit;
update s_db set perc = total / 4;
```

5. Computing result :

```
update s_db set result = 'pass' where Economics >= 35 and
Computer >= 35 and English >= 35 and Sanskrit >= 35;
```

Teacher's sign .....

S_ID	S_NAME	Economics	Computer	English	Sanskrit	Total	Percentage	Result
1	Krishna	67	35	20	90	212	35.33	Fail
2	Ram	80	40	90	70	280	46.6	Pass
3	Vittal	67	24	45	59	195	32.5	Fail

S_ID	S_Name
1	Krishna
2	Ram.
3	Vittal

Passed students
1

Failed Students
2

S_ID	S_NAME	Economics	Computer	English	Sanskrit	Total	Percentage	Result
1	Krishna	67	35	20	90	212	35.33	Fail
2	Ram	80	40	90	70	280	46.6	Pass
3	Vittal	67	24	45	59	195	32.5	Fail

COMPLETED

V.Good

7/3/24

update s-db set result = 'fail' where Economics < 35 or Computer < 35 or English < 35 or Sanskrit < 35

8. Displaying records:
select * from s-db;

7. Displaying id and name only:
select s-id, s-name from s-db;

8. Listing passed and failed:
select * from s-db where result = 'pass';
select * from s-db where result = 'fail';

9. Counting students passed / failed:
select count(*) from s-db where result = 'pass';
select count(*) from s-db where result = 'fail';

10. Selecting records with percentage >= 60%:
select * from s-db where perc >= 60;
→ Empty Set

11. Sorting by ID:
select * from s-db ~~where~~ order by s-id;

Teachers sign .....