



K.L.E. SOCIETY'S
P. C. JABIN SCIENCE COLLEGE
HUBBALLI
AUTONOMOUS

Semester I

B.Sc.

B.C.A.

M.Sc.

Answer Booklet No.

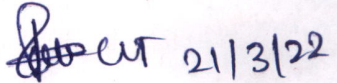
42485

Theory Semester End
Examination

March
April/May 2022

Nov./Dec. 20

Certified that the entries made by the candidate
are found to be correct.



Signature of the Room Supervisor with Date

Exam. Reg. No. 121MCS021

Class : B.Sc-I 'B' Subject : Computer Science Subject Code No. 124DS COIT

Paper :

12041019

PSTA



121MCS021

IMPORTANT INSTRUCTIONS TO CANDIDATES

- 1) On the cover page of answer book compulsorily mention your Register Number, Subject, Course Code and required information.
- 2) Don't write your name or mark any signs, such answer scripts shall not be assessed and punished.
- 3) Write your answer from 1st page and don't leave any blank pages and blank space in between.
- 4) Last page is meant for rough work and on completion put cross mark (x)
- 5) The candidates are informed strictly to write their answer only with black ink & write on both sides of the answers sheets.

IMPORTANT INSTRUCTIONS TO CANDIDATES

- 6) Please mention the Question number in the margin. Answer's without Question number & also with wrong question number shall not be valued.
- 7) The students are informed to take compulsorily the signature of the room supervisor with date on the answer book.
- 8) The candidate should be present 20 minutes before the commencement of the examination. After that no students will be allowed in the examination hall.
- 9) Use of any electronic gadgets in the examination hall is strictly prohibited.
- 10) After the last warning bell, no candidate is allowed to leave his/her seat.
- 11) Indulging in different ways and using different means that lead to malpractice is prohibited.
- 12) Don't fold the answers sheets & keep the answer sheets clean.

ಪ್ರಶ್ನೆ ಸಂಖ್ಯೆ
Question No.

ಉತ್ತರವನ್ನು ಇಲ್ಲಿಂದ ಆರಂಭಿಸಿ
START WRITING ANSWER FROM HERE BELOW

UNIT-II

3) a) Operators are the symbol, which operates on an variable (or) value. When a programmer wants to perform any kind of mathematical operation then he/she takes the help of the operator (Arithmetic operator). So operators are basically used to perform Arithmetic and logic operations.

Types of operators are;

- (i) Arithmetic operator
- (ii) Logical operator
- (iii) Assignment operator
- (iv) Bit-wise operator
- (v) Increment & decrement operator
- (vi) Conditional operator
- (vii) Special operator
- (viii) Relational operator

b) /* C - Program to find the entered no. is prime (or) not */

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int n, i;
    printf("Enter a number to check prime
           or not");
    scanf("%d", &n);
    int count = 0;
    for(i = 2; i < n; i++)
    {
        if(n % i == 0)
            count++;
    }
    if(count != 0)
        printf("The entered no. is not
               a prime number");
    else
        printf("The entered no. is a
               prime number");
    return 0;
    getch();
}
```

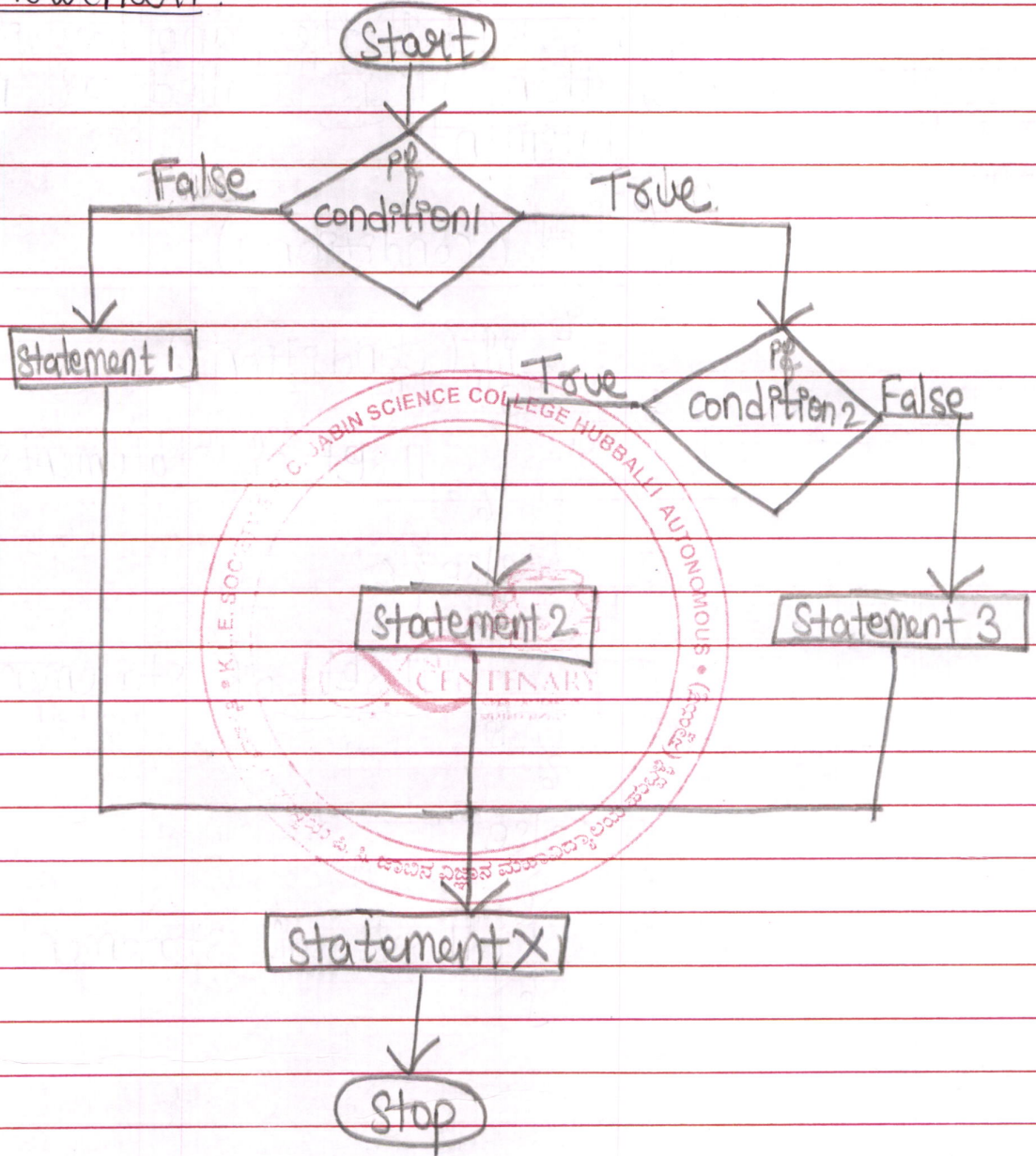
ಪ್ರಶ್ನೆ ಸಂಖ್ಯೆ
Question No.

c) If an if-else statement is present inside the block of the another if (or) else statement, then it is called as nested if-else statements.

Syntax :-

```
if (condition 1)
{
    if (condition 2)
    {
        // set of statements
    }
    else
    {
        // set of statements
    }
}
else
{
    // set of statements
}
```

Flowchart :-



Eg:-

/* C-Program to show working of
nested if-else statement */

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

```
#include <conio.h>
```

```
int main()
```

```
{
```

```
    int var1, var2;
```

```
    printf("Enter the value of var1.");
```

```
    scanf("%d", &var1);
```

```
    printf("Enter the value of var2.");
```

```
    scanf("%d", &var2);
```

```
    if (var1 != var2)
```

```
    {  
        printf("var1 is not equal to greater than var2);
```

```
        if (var1 > var2)
```

```
        {  
            printf("var1 is greater than var2);
```

```
        }  
    }  
    else
```

```
    {
```

```
        printf("var2 is greater than var1);
```

```
    }  
    else
```

```
    {
```

```
        printf("var1 is equal to var2);
```

```
    }  
    getch();
```

```
}
```

(OR)

4) a) In C-language there are 2 control statements and these are branching and looping. Branching decides, what action to take and looping decides, how many times, the particular action to be taken. Branching helps us to jump from one statement to the another statement without condition. Looping or Loop is a sequence of statements that repeats continuously until the stated condition becomes false. It has 2 parts in it, and they are (i) body of the loop and (ii) control statement.

b) /* C - Program to find the factorial of a given number */

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

```
#include <conio.h>
```

```
int main()
```

```
{
```

```
int n, i, fact = 1;
```

```
printf("enter a number to obtain the factorial");
```

```
scanf("%d", &n);
```

```
{
```

```
for (i = 0; i <= n; i++)
```

```
{
```


ಪ್ರಶ್ನೆ ಸಂಖ್ಯೆ
Question No.

```
    fact = fact * i
    printf("The factorial is %d", n, fact);
}
}
getch();
}
```

C) Formatted Input/Output functions :- 'C' provides 2 standard functions (scanf & printf) to perform formatted input and formatted output functions.

(i) scanf() :- It is used as the input for

These required a header to activate them and they also need a format specification for storing the parameters. Let us discuss them in the detail.

(i) scanf() :- It is used as the input formatted from the standard inputs. It helps in providing various options to the printf() statement.

Syntax :- scanf("format specifier, &data1, &data2");

Eg :- scanf("%d%c", &data1, &data2);

It reads and converts the parameter according to the format specification and it creates a memory space for it, which is represented by other arguments.

(ii) printf() :- It is used as the output formatted from the standard output. ~~It reads according to format~~ It needs a format specification.

Syntax :- `printf (format specifier, data1, data2)`
Eg :- `printf ("%d%c", data1, data2);`

So these are about the formatted Input/output functions in 'C'.

UNIT-I

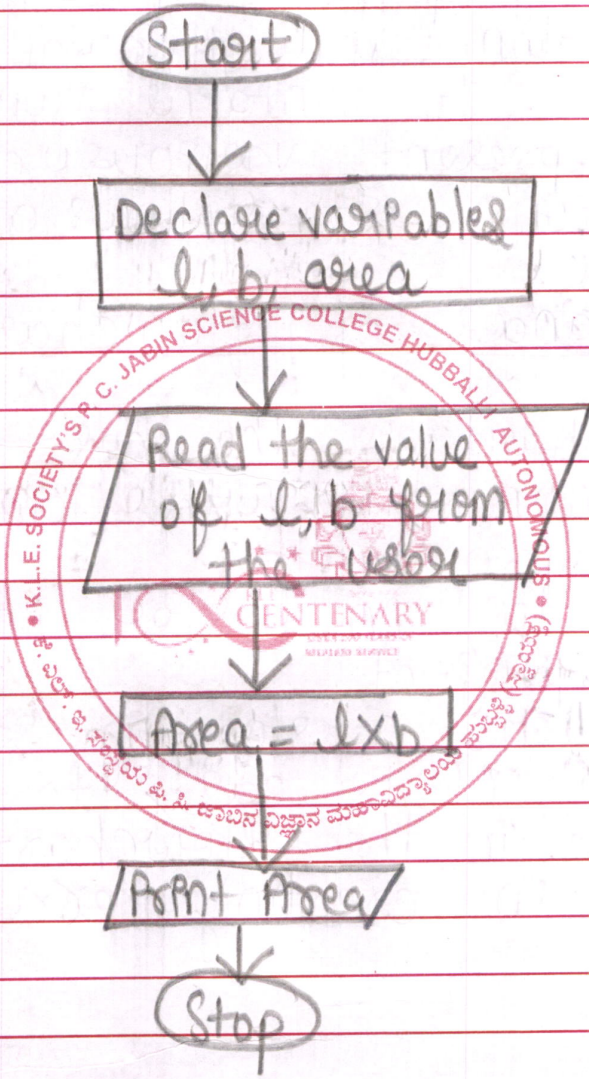
2) a) A flowchart is the most widely used graphical representation of an algorithm and procedural design work flows. It contains various symbols to represent various operations and decisions. It follows a sequential order.

• Advantages of flowchart :-

- (i) Flowchart is the best way of communicating the logic flow of the program.
- (ii) It helps in faster implementation of the project.
- (iii) As it is a step-by-step procedure, hence it is easy to debug in any -where in the flowchart.
- (iv) Helps in creating error-free programs.

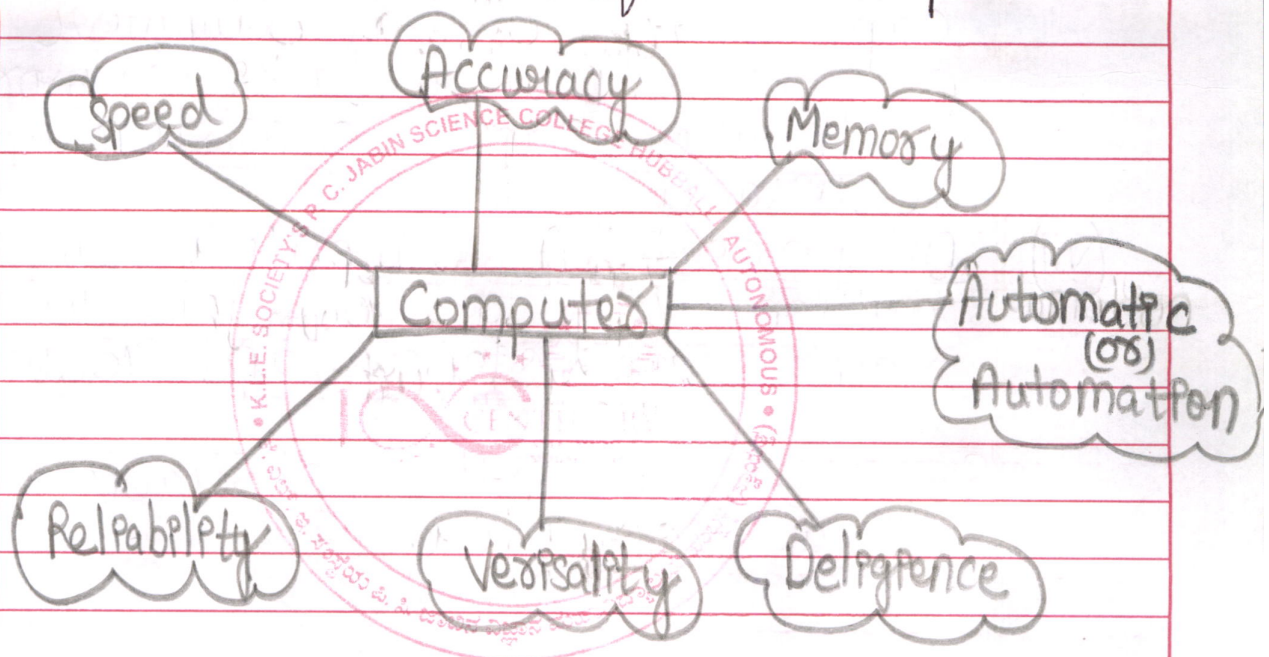
ಪ್ರಶ್ನೆ ಸಂಖ್ಯೆ
Question No.

b) Flowchart:- To find area of the rectangle.



c) A computer is an electronic device which can manipulate data (or) information. It has the ability to store, retrieve and process the data.

Characteristics of a Computer:-



Let us discuss them in detail:-

- (i) Speed - A computer can work with much higher speed when compared to humans. A computer can process millions of instructions per second.
- (ii) Accuracy - A computer can perform calculations with 100% accuracy.
- (iii) Memory - A computer has an built-in memory called the primary memory where it stores all the data.

ಪ್ರಶ್ನೆ ಸಂಖ್ಯೆ
Question No.

- (iv) Automatic (or) Automation - A computer works automatically.
- (v) Reliability - A computer is reliable as it gives the same result for similar set of data.
- (vi) Versatility:- It is the capacity (or) capability of a computer to perform many tasks (or) works with the same consistency and accuracy.
- (vii) Diligence A computer can process million's of tasks (or) jobs with the same consistency and accuracy.

So, these are the characteristics of a computer.

UNIT-III

5) a) An array is a collection of data ~~(or)~~ data types stored at contiguous memory locations.
Advantages of array:-

- (i) Code optimization :- Code is easier to write.
- (ii) Ease of sorting :- We can easily sort the elements ~~(or)~~ data of an array by writing a few lines of code only.
- (iii) Random access :- We can randomly access the elements of the array by just using their index numbers.
- (iv) Ease of traversing - By just using the 'for' loop we can retrieve the elements ~~(or)~~ data of the array.

b) A string is a collection of characters enclosed under the double quoted.
Eg:- "Computer".

The following are the string functions.

→ P.T.O.

ಪ್ರಶ್ನೆ ಸಂಖ್ಯೆ
Question No.

String functions

Description

- | | | |
|-------|--|--|
| (i) | strlen (first_string
second_string) | It returns the length of the string. |
| (ii) | strcat (first_string
second_string) | It concatenates (or) joins the 1 st string with the 2 nd string. |
| (iii) | strcmp (first_string
second_string) | It compares the 1 st string with 2 nd string. |
| (iv) | strrev (string) | It returns the reverse of the string. |
| (v) | strupr (string) | It returns the string in uppercase. |
| (vi) | strlower (string) | It returns the string in lowercase. |
| (vii) | strcpy (str) | It copies the string str2 into string str1. |

So these are the string functions.

c) Call by Value / Pass by Value :- A copy of data is made and stored by the name of the arguments in the original data (or) calling function. Hence any changes made to the parameter does not affect the calling function (or) original data. This is called as call by Value / Pass by value.

Eg:-

```
/* C-Program to represent call by value */  
#include <stdio.h>  
#include <conio.h>  
void swap(int a, int b);  
int main()  
{  
    int a, b;  
    printf("Enter any 2 numbers");  
    scanf("%d %d", &a, &b);  
    swap(a, b);  
    printf("The main function is  
           a = %d, b = %d\n", a, b);  
    getch();  
}  
void swap(int a, int b);  
{  
    int temp;
```

ಪ್ರಶ್ನೆ ಸಂಖ್ಯೆ
Question No.

```
temp = a;  
a = b;  
b = temp;  
printf("The calling function is  
a = %d, b = %d", a, b);  
}
```

UNIT - IV

7) a) 'C-language' uses top-down approach to solve a problem, in this approach it starts with a high-level design and ends with a low level implementation. That is why 'C' is called as top down approach.

a) As 'C' - is a structural language, hence it uses top-down approach to solve a problem, in this approach it starts with a high level design and ends with a low level implementation. That is why 'C' is called as top-down approach.

ಪ್ರಶ್ನೆ ಸಂಖ್ಯೆ
Question No.

b) /* C - Program to swap two numbers */

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

```
#include <conio.h>
```

```
int main()
```

```
{
```

```
    int a, b, temp;
```

```
    printf("Enter any two numbers");
```

```
    scanf("%d %d", &a, &b);
```

```
    printf("Before Swapping a=%d,  
           b=%d\n", a, b);
```

```
    temp = a;
```

```
    a = b;
```

```
    b = temp;
```

```
    printf("After Swapping a=%d,  
           b=%d\n", a, b);
```

```
    getch();
```

```
}
```

c) Problem Solving in C :-

Problem solving is a sequential process of analysing the information and generating an appropriate response option. While working with the computer, one has to undergo through the following steps (or) stages to solve a problem, and these are;

- (i) Understanding the problem.
- (ii) Analysing the problem.
- (iii) Developing a solution.
- (iv) Coding and implementation.

Now let us discuss them in detail;

(i) Understanding the problem :- Here, we try to understand the problem to be solved in totally. Before moving on to the next step we must be clear with the objectives of the given problem.

(ii) Analysing the problem :- After understanding the problem to be solved, we find out different methods to solve this problem, and evaluate each of these methods.

(iii) Developing the solution :- Here, we have a sequence of operations, that was formed as the result of the ^{an} ~~an~~ state. Now we apply these operations in a certain order to the problem in the consideration, ~~int~~ in order to get the desired output. And also find ways of applying it to the problem.

(iv) Coding and implementation :- The last stage in the problem solving is conversion of the steps into an language, that the computer can understand. Here each statement is converted into an computer language that is chosen for the implementation.

So, these are the stages of problem solving in 'C'.

ಪ್ರಶ್ನೆ ಸಂಖ್ಯೆ
Question No.

UNIT-III (OR)

6) c) /*C - Program to find the largest element of an array using 1-D array*/

```
#include <stdio.h>
#include <conio.h>
int max (int arr[], int size);
int main()
```

```
{
```

```
int x, p;
```

```
x = arr[0]
```

```
for (p = 0; p < size; p++)
```

```
{
```

```
if (arr[p] < x)
```

```
x = arr[p]
```

```
}
```

```
return x;
```

```
}
```

```
int main()
```

```
{
```

```
int arr[5];
```

```
int m, i;
```

```
printf("enter the array elements");
```

```
for (i = 0; i < 5; i++)
```

```
scanf("%d", &arr[i]);
```

```
m = max(arr, 5);
```

```
printf("The max in the array is %d", m);
```

```
getch();
```

```
}
```

a) We can initialize an array simply by using its index number. Array can also be initialize at the time of declaration.

Syntax :- return_type $\text{array_name}(\text{array_size})$

Eg :- $\text{int marks}[5] = \{10, 20, 30, 40, 50\}$

b) User defined functions :- The functions that are declared by the user for the program requirement is called as user-defined functions.

Syntax :- return_type $\text{function_name}(\text{argument_list})$

Set of statements

(i) return type :- It contains all the data types (int, float, char) etc.

(ii) Function name :- It is a name given to the function for the ease of using in the program.

However, it is suggested to give a meaningful name, for easy understanding and use.

(iii) Argument - list :- This list contains all the variables along with their data types. The argument list is basically an input for the functions.

(iv) Block of Code - It is a set of 'C'-statements which will be executed when a function call is made.

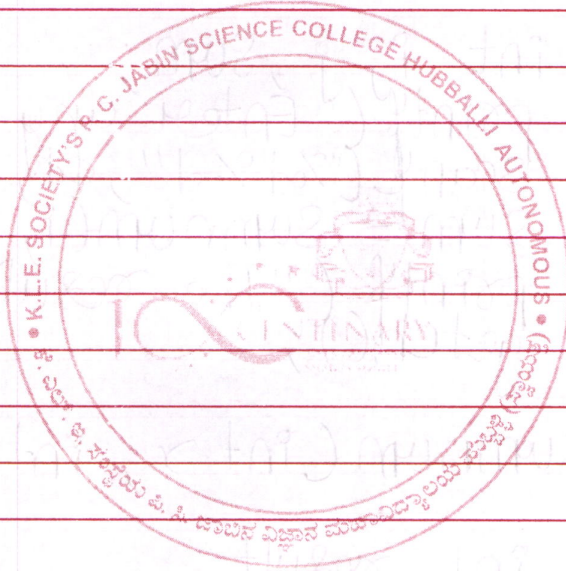
ಕ್ರಮ ಸಂಖ್ಯೆ
Question No.

Eg:- /* C-Program to use show user-defined function */

```
#include <stdio.h>
#include <conio.h>
int sumnum(int x, int y);
int main()
{
    int i, j, sum;
    printf("Enter any two numbers");
    scanf("%d %d", &i, &j);
    sum = sumnum(i, j);
    printf("The result is %d", sum);
    getch();
}

int sumnum(int x, int y);
{
    int result;
    result = x + y;
    return result;
}
```

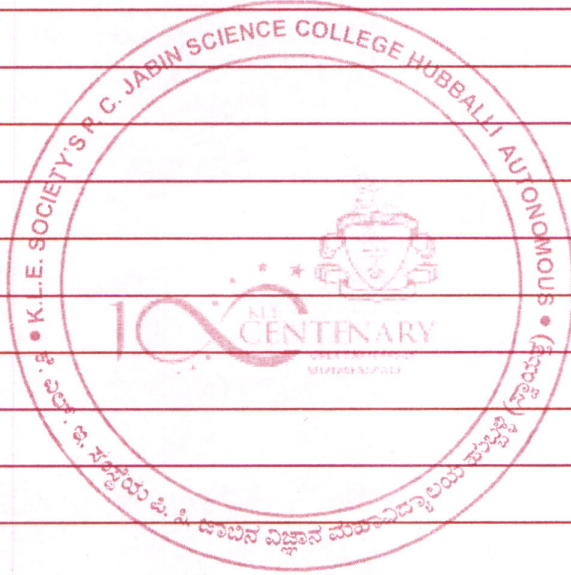
ಪ್ರಶ್ನೆ ಸಂಖ್ಯೆ
Question No.



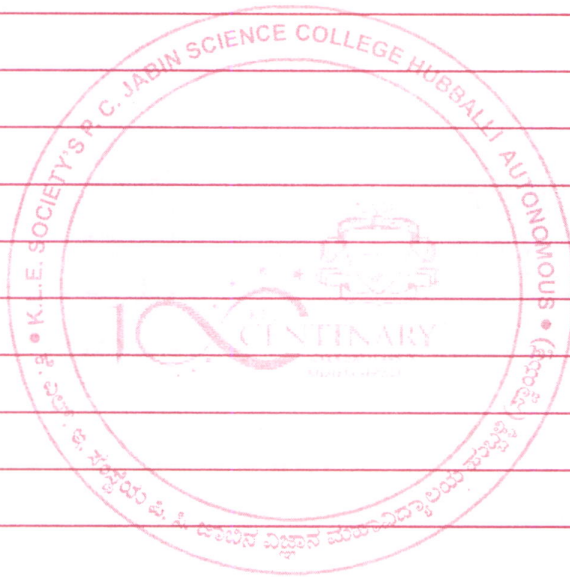
ಪ್ರಶ್ನೆ ಸಂಖ್ಯೆ
Question No.



ಪ್ರಶ್ನೆ ಸಂಖ್ಯೆ
Question No.



ಪ್ರಶ್ನೆ ಸಂಖ್ಯೆ
Question No.



ಪ್ರಶ್ನೆ ಸಂಖ್ಯೆ
Question No.



ಪ್ರಶ್ನೆ ಸಂಖ್ಯೆ
Question No.



ಪ್ರಶ್ನೆ ಸಂಖ್ಯೆ
Question No.

