

ಪರೀಕ್ಷಾ ಕೊಠಡಿಯಲ್ಲಿ ಅಭ್ಯರ್ಥಿಗಳು ಅನುಸರಿಸಬೇಕಾದ ಸೂಚನೆಗಳು

INSTRUCTIONS TO CANDIDATES (at the examination hall)

1. ಪ್ರವೇಶ ಪತ್ರದ ಹಿಂಭಾಗದಲ್ಲಿ ಮುದ್ರಿಸಿರುವ ಸೂಚನೆಗಳನ್ನು ತಪ್ಪದೇ ಅನುಸರಿಸಬೇಕು.  
Please follow the instructions given in the Admission Ticket Carefully.
2. ಉತ್ತರಿಸುವ ಮುನ್ನ ಉತ್ತರ ಪತ್ರಿಕೆಯಲ್ಲಿ 40 ಪುಟಗಳು ಇದೆಯೇ ಎಂಬುದನ್ನು ಖಚಿತ ಪಡಿಸಿಕೊಳ್ಳಿ, ಸರಿಯಾಗಿ ಇಲ್ಲದಿದ್ದಲ್ಲಿ, ಕೊಠಡಿಯ ಮೇಲ್ವಿಚಾರಕರಿಂದ ಮತ್ತೊಂದು ಉತ್ತರ ಪತ್ರಿಕೆಯನ್ನು ಪಡೆಯಬೇಕು.

Before answering the questions, ensure that the answer book contains 40 pages. In case it is defective, request the invigilator to issue a new answer book.

3. ಪ್ರವೇಶ ಪತ್ರಿಕೆ ಮತ್ತು ಪ್ರಶ್ನೆ ಪತ್ರಿಕೆಯಲ್ಲಿ ನಮೂದಿಸಿರುವ ಪ್ರಶ್ನೆ ಪತ್ರಿಕೆಯ ಕೋಡ್ ಸರಿಯಿದೆಯೆ ಎಂಬುದನ್ನು ಖಚಿತಪಡಿಸಿಕೊಳ್ಳಿ.

Ensure that the QP code on Admission Ticket and actual Question Paper are identical.

4. ಅಭ್ಯರ್ಥಿಗಳು ಪರೀಕ್ಷೆಗೆ ಸಂಬಂಧ ಪಟ್ಟ ವಿಷಯಗಳಾದ ಡಿಗ್ರಿ, ವಿಷಯ, ಹೆಸರು, ರಿಜಿಸ್ಟ್ರ್ ನಂಬರ್, ಪರೀಕ್ಷಾ ದಿನಾಂಕ, ಸ್ಟುಡೆಂಟ್ ಐ.ಡಿ. ಹಾಗೂ ಪ್ರಶ್ನೆ ಪತ್ರಿಕೆಯ ಕೋಡ್‌ನ್ನು (ಕ್ಯೂ ಪಿ ಕೋಡ್) 1 ನೆಯ ಪುಟದಲ್ಲಿರುವ ಭಾಗ-1 ರಲ್ಲಿ ಬರೆಯಬೇಕು.

Candidates are instructed to write information such as Degree, Exam, Subject, Candidate name, **Register No.**, Exam date, **Student ID and Question paper code (QP Code)** in Part-I.

5. ಸ್ಟುಡೆಂಟ್ ಐ.ಡಿ. ಹಾಗೂ ಪ್ರಶ್ನೆ ಪತ್ರಿಕೆಯ ಕೋಡ್‌ನ್ನು (ಕ್ಯೂ ಪಿ ಕೋಡ್) ನಿಗದಿ ಪಡಿಸಿದ ಬಾಕ್ಸ್‌ನಲ್ಲಿ ಬರೆಯಬೇಕು ಮತ್ತು ಕೆಳಗೆ ಇರುವ ಸಂಬಂಧಪಟ್ಟ ಸರ್ಕಲ್‌ಗಳನ್ನು ಶೇಡ್ ಮಾಡಬೇಕು. ಇದರ ನಮೂನೆಯನ್ನು 2 ನೆಯ ಪುಟದಲ್ಲಿ ತೋರಿಸಲಾಗಿದೆ.

Student ID and QP code are to be written in the designated boxes. Appropriate ovals below these boxes are to be shaded as shown in the sample in Page 2

6. ನಕ್ಷೆ ಅಥವಾ ಗ್ರಾಫ್ ಹಾಳೆಗಳನ್ನು ಉಪಯೋಗಿಸಿದಲ್ಲಿ ಅವುಗಳ ಮೇಲೆ ರಿಜಿಸ್ಟ್ರ್ ನಂಬರ್ ಹಾಗೂ ಸ್ಟುಡೆಂಟ್ ಐ.ಡಿ. ಬರೆಯಬಾರದು.

Do not write Register No. / Student ID on map / graph sheet, if used.

7. ಉತ್ತರ ಪತ್ರಿಕೆಯ ಕ್ರಮಸಂಖ್ಯೆಯನ್ನು ಕೊಠಡಿಯ ಮೇಲ್ವಿಚಾರಕರ ದಿನಚರಿಯಲ್ಲಿ ಬರೆದು ಪೂರ್ಣ ಸಹಿ ಮಾಡಬೇಕು.

Candidates shall write the serial number of the Answer books and affix their signature, in token of their receiving the answer book in the invigilator's Dairy.

8. ಉತ್ತರಗಳನ್ನು ನೀಲಿ ಅಥವಾ ಕಪ್ಪುಹಿಂಡು ಹೆಸನ್ನು ಮಾತ್ರ ಉಪಯೋಗಿಸಿ ಬರೆಯಬೇಕು. ಉತ್ತರ ಪತ್ರಿಕೆಗಳ ಜೆರಾಕ್ಸ್ ಪ್ರತಿ ಇಚ್ಛಿಸುವವರು ಉತ್ತರಗಳನ್ನು ಕಪ್ಪುಹಿಂಡಿನಲ್ಲಿ ಮಾತ್ರ ಬರೆಯಬೇಕು.

Answers must be written in black or blue ink. Those who desire to obtain xerox copy of their answer books must write in black ink.

9. ಉತ್ತರ ಪತ್ರಿಕೆಯಲ್ಲಿ ಖಾಲಿ ಉಳಿಯುವ ಎಲ್ಲಾ ಹಾಳೆಗಳ ಮೇಲೆ ಉದ್ದನೆಯ ಗೆರೆ ಎಳೆಯಬೇಕು. Score off all the blank pages left in the answer books at the end of the examination.

10. ಪರೀಕ್ಷಾ ಅವಧಿ ಮುಗಿಯುವ ಮುನ್ನ ಪರೀಕ್ಷಾ ಕೇಂದ್ರದಿಂದ ಹೊರಗೆ ಹೋದಲ್ಲಿ ಪ್ರಶ್ನೆ ಪತ್ರಿಕೆಯನ್ನು ಕೊಠಡಿಯ ಮೇಲ್ವಿಚಾರಕರಿಗೆ ಕೊಡಬೇಕು.

Candidate is required to handover the question paper to the room invigilator in case he / she leaves the examination hall before the closing of

11. ಉತ್ತರಗಳ:

ಮೇಲ್ವಿಚಾರ

When ca

sition the

12. ಉತ್ತರ ಪತ್ರಿಕೆಯನ್ನು

ಓಂ ಎಂದಾಗಲಿ,

ನಮಃ ಶಿವಾಯ,

ಚುಕ್ಕೆಗಳು ಇತ್ಯಾದಿಗಳನ್ನು ಬರೆಯಬಾರದು.

Code marks revealing identity such as "Om", "Shri", "Namah-Shivaya" etc., should not be written any where in the answer books.

13. ಮೊಬೈಲ್ ಫೋನ್‌ಗಳನ್ನು ಹಾಗೂ ಇತರೆ ಎಲೆಕ್ಟ್ರಾನಿಕ್ಸ್ ಉಪಕರಣಗಳನ್ನು ಪರೀಕ್ಷಾ ಕೊಠಡಿಯೊಳಗೆ ತರಬಾರದು.

Do not bring Mobile Phones and other electronic equipments inside the Examination Hall.

14. ಉತ್ತರ ಪತ್ರಿಕೆಯಿಂದ ಯಾವುದೇ ಪುಟವನ್ನು ಹರಿಯಬಾರದು ಅಥವಾ ತೆಗೆಯಬಾರದು.

Do not tear or remove any sheet from the Answer Booklet



QP: 54013 Pkt : 0001 Sl.: 02

PROG. FUNDAMENT. &

ಗೊನೆಗೆ) ಕೊಠಡಿಯ  
ಬೇಕು.

Shall continue to  
book.

ಎಂದಾಗಲಿ,

ನಮಃ ಶಿವಾಯ,

ಚುಕ್ಕೆಗಳು ಇತ್ಯಾದಿಗಳನ್ನು ಬರೆಯಬಾರದು.

Code marks revealing identity such as "Om", "Shri", "Namah-Shivaya" etc., should not be written any where in the answer books.

ಮೊಬೈಲ್ ಫೋನ್‌ಗಳನ್ನು ಹಾಗೂ ಇತರೆ ಎಲೆಕ್ಟ್ರಾನಿಕ್ಸ್ ಉಪಕರಣಗಳನ್ನು ಪರೀಕ್ಷಾ ಕೊಠಡಿಯೊಳಗೆ ತರಬಾರದು.

Do not bring Mobile Phones and other electronic equipments inside the Examination Hall.

ಉತ್ತರ ಪತ್ರಿಕೆಯಿಂದ ಯಾವುದೇ ಪುಟವನ್ನು ಹರಿಯಬಾರದು ಅಥವಾ ತೆಗೆಯಬಾರದು.

Do not tear or remove any sheet from the Answer Booklet

ಕುವೆಂಪು ವಿಶ್ವವಿದ್ಯಾಲಯ



KUVEMPU UNIVERSITY

Part - IV

Qn.	Marks	Total Marks	Qn.	Marks	Total Marks
1			13		
2			14		
3			15		
4			16		
5			17		
6			18		
7			19		
8			20		
9			21		
10			22		
11			23		
12			24		
Sub Total			Sub Total		

To be filled by Examiner (Third Valuation)

Examiner Code

Date

Signature of the Examiner

Reviewer Code

Signature of the Reviewer



Grand Total in words

Grand Total

DO NOT WRITE





## Part - A

Ans ⑥:- ~~Constant~~  $\rightarrow$

Ans ①:- Assembly language  $\rightarrow$  A programming language that is once removed from a computer machine language. Machine languages consist entirely of numbers and are almost impossible for humans to read and write. Assembly languages have the same structure and set of commands as machine language, but they enable a programmer to use names instead of numbers.

Ans ②:- Program to generate Fibonacci series:-

```
Using namespace std;
main()
```

```
{
```

```
int n, c, first = 0, second = 1, next;
cout << "Enter the number of terms of
```



```
Fibonacci series you want" << endl;  
Cin >> n;
```

```
Cout << "First " << n << " terms of  
Fibonacci Series are :- " << endl;
```

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

```
{  
    if ( c <= 1)
```

```
        next = c;
```

```
    else
```

```
{
```

```
        next = first + second;
```

```
        first = second;
```

```
        second = next;
```

```
    }
```

```
    Cout << next << endl;
```

```
    return 0;
```

```
}
```





Ans ③ :- Program :-

```
#include <stdio.h>
#include <conio.h>
void main()
{
    float c, f;
    clrscr();

    /* To convert degree to Fahrenheit */
    printf("Enter the temperature in Degree.");
    scanf("%f", &c);
    f = 1.8 * c + 32;
    printf("1n1t %.2f Degree = %.2f Fahrenheit".c, f);

    /* To convert Fahrenheit to Degree */
    printf("1n1n Enter the temperature in Fahrenheit.");
    scanf("%f", &f);
    c = (f - 32) / 1.8;
    printf("1n1t %.2f Fahrenheit = %.2f Degree", f, c);
    getch();
}
```





Ans (4):- Machine level lang. | High level language

① Machine language  
uses binary numbers  
or codes

① High level lan  
use keywords  
similar to English  
and are easier to  
write.

② It is a low level  
language and  
is machine dependant.

② Machine dependant  
or while  
② High level lan  
are not dependant

Ans (5):- Accessing Data using of a particular  
field of a structure:-

All of the structures discussed up  
to this point have elements that  
can only be referenced using fixed  
field names. Another means of  
accessing structures is to use  
dynamic field names. These names  
express the field as a variable





expression that MATLAB evaluates at run-time. The dot-parentheses syntax shown here makes expression a dynamic field name.

struct\_name.(expression).

Ans(6):- Constant

→ Constant is a identifier with an associated value which cannot be altered by the program during normal execution - the value is constant.

Ans(7):- Use of function proto typing →

In computer programming, a function prototype or function interface is a declaration of a function that specifies the function's name and type signature (arity, parameter types, and return type), but omits the function body.  
~~The term is pass~~





Ans 8) Union  $\rightarrow$  A union is a special data type available in C that enables you to store different data types in the same memory location. You can define a union with many members, but only one member can contain a value at any given time. Unions provide an efficient way of using that same memory location for multipurpose.





## Part-B

Ans(9) (a) The basic structure of C program:

- Documentation Section
- Link Section
- Definition Section
- Global Declaration Section
- main() function section
- Subprogram Section

(i) Documentation Section

→ This section consists of comment lines which include the name of programmer, the author and other details like time and date of writing the program. It also helps anyone to get an overview of the program.

(ii) Link Section

→ The link section consists of the header files of the functions that are used in the program. It provides instructions to the compiler to link





function from the system library.

(iii) Definition section  $\rightarrow$  All the symbolic constants are written in definition section. Macros are known as symbolic constants.

(iv) Global declaration section  $\rightarrow$  The global variable that can be used anywhere in the program are declared in global declaration section.

(v) main() function section  $\rightarrow$  It is necessary to have one main() function section in every C program. This section contains two parts, declaration and executable part. The declaration part declares all the variables that are used in executable part.





(vi) Subprogram section  $\rightarrow$  This section consists contains all the user defined functions that are used to perform a specific task. These user defined function are called in the main function.

Ans (b) Some of basic rules need to follow while constructing flow chart are:-

- (i) All boxes of the flowchart are connected with arrows.
- (ii) Flowchart Symbols have an entry point on the top of the symbol with no other entry points. The exit point for all flowchart symbols is on the bottom except for the Decision symbol.
- (iii) The decision symbol has two exit points; these can be on the sides or the bottom and one side.
- (iv) Generally a flowchart will flow from top to bottom. However, an upward flow can be shown as long as it does not exceed 3 symbols.





(v) Connectors are used to connect boxes in the flowchart. e.g. are

- From one page to another page.
- From the bottom of the page to top of the same page.

• An upward flow of more than 3 symbols.

(vi) Subroutines and Interrupt programs have their own and independent flowcharts.

(vii) All flowcharts start with a start or predefined process symbol.

(viii) All flowchart end with a terminal or a contention loop.



Ans(12) (a) Bit wise operator  $\rightarrow$  The Bit wise operator supported by C language are listed in the following table. Assume variable A holds 60 and variable B holds 13, then:

Operator	Description	Example.
&	Binary <del>And</del> AND operator (copies a bit to the result if it exists in both operands)	(A & B) will give 12 which is 0000 1100
	Binary OR operator (copies a bit if it exists in either operand.)	(A   B) will give 61 which is 0011 1101
^	Binary XOR operator (copies the bit if it is set in one operand but not both.)	(A ^ B) will give 49 which is 0011 0001





~

Binary ones complement operator is unary and has the effect of 'flipping' bits.

(A) ~  
-61  
1100  
2's complement  
form  
a signed  
number

1100 (44 A)

2's complement  
0011 0000

Binary left shift operator. The left operands value is moved left by the number of bits specified by the right operand.

A <<  
give 2  
which  
1111 00

1100 (8 A)

1's complement

>>

(011100

Binary right shift operator. The left operands value is moved right by the number of bits specified by the right operands.

A >>2  
give 5  
is  
0000.

1100 (8 A)

1's complement

1000 1100





(Give  
which is  
in  
plement  
to  
Binary

will  
0

is  
20

will  
which

11)

Example.

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

```
main()
```

```
{
```

```
    unsigned int a = 60; /* 60 = 0011 1100 */
```

```
    unsigned int b = 13; /* 13 = 0000 1101 */
```

```
    C = a & b; /* 12 = 0000 1100 */
```

```
    printf ("Line 1 - Value of C is %d\n", C);
```

```
    C = a / b; /* 61 = 0011 1101 */
```

```
    printf ("Line 2 - Value of C is %d\n", C);
```

```
    C = a ^ b; /* 49 = 0011 0001 */
```

```
    printf ("Line 3 - Value of C is %d\n", C);
```

```
    B
```

```
    C = ~a; /* -61 = 1100 0011 */
```

```
    printf ("Line 4 - Value of C is %d\n", C);
```

```
    C = a << 2; /* 240 = 1111 0000 */
```

```
    printf ("Line 5 - Value of C is %d\n", C);
```

```
    C = a >> 2; /* 15 = 0000 1111 */
```

```
    printf ("Line 6 - Value of C is %d\n", C);
```

```
}
```





When you compile and execute above program:-

Output :-

Line 1 - Value of C is 12

Line 2 - Value of C is 61

Line 3 - Value of C is 49

Line 4 - Value of C is -61

Line 5 - Value of C is 240

Line 6 - Value of C is 15

(G) The Different Storage classes in C

- auto
- register
- static
- extern

(i) The auto storage class  $\rightarrow$  The auto

Storage class is the default storage class for all local variables.

```
{  
    int count;
```





he

```

auto int month;
}

```

The e.g above definition define two variable with the same storage class, auto can only be used within functions. i.e, local variables.

(ii) Register storage class

→ The register storage class is used to define local variable that should be stored in a register instead of RAM. This means that the variable has a no maximum size equal to the register size and can't have the unary '&' operator applied to it.

2.   
 (i)   
 {

```

register int miles;
}

```

(iii) The Static storage class → It instruct the compiler to keep a local variable in existence during the life-time of the program instead of creating





destroying it each time it comes in and goes out of scope. ~~Therefore~~ ~~making~~ f.

(iv) The extern storage class → It is used to give a reference of a global variable that is visible to All the program.

Ans (13) (a) Program to copy the contents of one string to another :-

```
#include <stdio.h>
#include <conio.h>
void main ()
{
    int i = 0;
    char S1[10], S2[10];
    clrscr();
    printf("In enter the string 1 \n");
    gets(S1);
    printf("In enter the string 2 \n");
    gets(S2);
    printf("In string 1 :- %s", S1);
```





```
printf("In String 2 :- %s", S2);  
while(S1[i] != NULL)  
{  
    S2[i] = S1[i];  
    i++;  
}  
S2[i] = '\0';  
printf("In After copy the String 2 = %s", S2);  
getch();  
}
```

## (b) Preprocessor Statements

→ The C preprocessor is the macro preprocessor for the C computer programming language. The preprocessor provides the ability for the inclusion of header files, macro expansions, conditional compilation, and line control.

In many C implementations, it is a separate program invoked by the compiler as the first part of translation.





The language of preprocessor directives is only weakly related to the grammar of C, and so is sometimes used to process other kinds of text files.

Ans 16:-

(c) Conditional compilation  $\rightarrow$  In computer programming, conditional compilation is a compilation implementing method which allows the compiler to produce differences in the executable program controlled by parameters that are provided during compilation. This technique is commonly used when these differences are needed to run the software on different platforms, or with different versions of required libraries or hardware.





(b) Linked list  $\rightarrow$

In Computer Science, a linked list is a data structure consisting of a group of nodes which together represent a sequence. Under the simplest form, each node is composed of data and a reference to the next node in the sequence; more complex variants add additional links. This structure allows efficient insertion or removal of elements from any position in the sequence.

(c) Getch function  $\rightarrow$

"The getch = get character. This function is used to get a single character input from the user" during execution of program. It also forces to wait the output to stay on screen until any key pressed from keyboard.





## Syntax:

- for value input: variable name = `getch()`;
- For screen holding at the end of program: `getch()`;

Ans (10) (a) Arithmetic operators →

Assume  $A = 10$  &  $B = 20$

Operator	Description	Example
+	Add two operands	$A + B$ will give
-	Subtracts second operand from the first	$A - B$ will give
*	Multiplies both operands	$A * B$ will
/	Divides numerator by denominator	$B / A$ will
%	Modulus operator and remainder of after an integer division	$B \% A$ will





$++$  Increment operator increase integer value by one  $A++$  will give 11

$--$  Decrement operator decreases integer value by one  $A--$  will give 9.

Logic operators  $\rightarrow$  Assume  $A=1$  &  $B=0$

Operator	Description	Example
$\&\&$	Called Logical AND operator. If both the operands are non-zero, the condition becomes true.	$(A \&\& B)$ is false
$  $	Called logical OR operator. If any of the two operands is non-zero, then condition become true.	$(A    B)$ is true.





! called Logical NOT ! ( $A \& B$ )  
operator. Use to is true.  
reverses the logical  
State of its operand.  
If a condition is true  
then logical NOT operator  
will make false

(G) C program to illustrate use unary  
arithmetic operators :-

```
main()
```

```
{
```

```
int a, b, c, d;
```

```
a = 15;
```

```
b = 10;
```

```
c = ++a - b;
```

```
printf ("a = %d b = %d c =  
a, b, c);
```

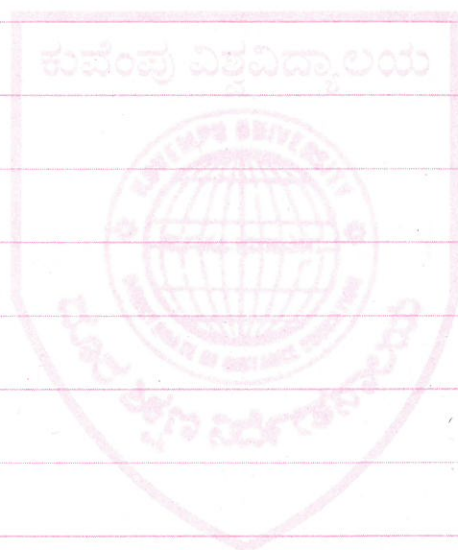
```
d = b++ + a;
```





4

```
printf (" a / b = %d \n", a / b);  
printf (" a % % b = %d \n", a % b);  
printf (" a * = b = %d \n", a * = b);  
printf (" %d \n", (c > d) ? 1 : 0);  
printf (" %d \n", (c < d) ? 1 : 0);  
?
```



d \n",