**B.Sc.** Mathematics

S5MEL1A-Fundamentals of Programming in C-Fifth Semester

Time: 3 Hrs Max. Marks: 75

**Part – A Answer all the Questions 10 X 2 Marks = 20 Marks**

|  |  |
| --- | --- |
|  | Write the series of steps in executing C program.   * Creating the program * Compilling the program * Linking the program with functions that are needed from the C library * Executing the program |
|  | C நிரலை இயக்குவதற்கான தொடர் படிகளை எழுதவும். |
|  | What are the relational operators used in C?   * Equal to. * Not equal to. * Less than. * Greater than. * Less than or equal to. * Greater than or equal to. |
|  | C இல் பயன்படுத்தப்படும் Relational ஆபரேட்டர்கள் யாவை? |
|  | Write the value of expression 9-12/4+2\*3-1 which will be executed in C Language.  #icnlude<stdio.h>  int main(){  int sum;  sum=(9-12)/4+(2\*3)-1;  printf(“%d”,sum);  } |
|  | சி மொழியில் செயல்படுத்தப்படும் வெளிப்பாடு 9-12/4+2\*3-1 மதிப்பை எழுதவும். |
|  | Write any two mathematical function used in C.   * Square Root. To find the square root of a number, use the sqrt() function: ... * Round a Number. The ceil() function rounds a number upwards to its nearest integer, and the floor() method rounds a number downwards to its nearest integer, and returns the result: ... * Power. |
|  | C இல் பயன்படுத்தப்படும் ஏதேனும் இரண்டு கணித செயல்பாட்டை எழுதவும். |
|  | Write the syntax of simple IF statement in C.  if (condition is true){     Statement (s)  } |
|  | எளிய IF அறிக்கையின் தொடரியல் C இல் எழுதவும். |
|  | Write any two differences between While statement and Do statement.   * While loop checks the condition first and then executes the statement(s), whereas do while loop will execute the statement(s) at least once, then the condition is checked. * While loop is entry controlled loop, whereas do while is exit controlled loop. |
|  | While statement மற்றும் Do statement ஆகியவற்றுக்கு இடையே ஏதேனும் இரண்டு வித்தியாசங்களை எழுதவும். |
|  | Explain compile time initialization of one dimensional array with an example.  Compile-Time initialization is also known as **static-initialization**. In this, array elements are initialized when we declare the array implicitly.  **Syntax:**  <data\_type> <array\_name> [array\_size]={list of elements};  **Example:**  int nums[5] = {0, 1, 2, 3, 4}; |
|  | ஒரு பரிமாண வரிசையின் தொகுக்கும் நேர துவக்கத்தை எடுத்துக்காட்டுடன் விளக்கவும். |
|  | Define the term “Recursion”.  Recursion is **the technique of making a function call itself**. This technique provides a way to break complicated problems down into simple problems which are easier to solve. Recursion may be a bit difficult to understand. |
|  | "Recursion " என்ற சொல்லை வரையறுக்கவும். |
|  | How to declare a structure variable in C?  **struct**  {  **char** name[50];  **char** street[100];  **char** city[50];  **char** state[20];  **int** pin;  };  A structure variable can either be declared with structure declaration or as a separate declaration like basic types. |
|  | C இல் ஒரு Structure மாறியை எவ்வாறு அறிவிப்பது? |
|  | Write the syntax for declaring and opening file in C program.  To create a file in a 'C' program following syntax is used, **FILE \*fp; fp = fopen ("file\_name", "mode");** In the above syntax, the file is a data structure which is defined in the standard library. fopen is a standard function which is used to open a file. |
|  | C நிரலில் கோப்பை அறிவிப்பதற்கும் திறப்பதற்கும் தொடரியல் எழுதவும். |

**Part – B Answer all the Questions 5 X 5 Marks = 25 Marks**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | a) | Explain the basic structure of C program.   |  |  | | --- | --- | | Documentation | Consists of the description of the program, programmer's name, and creation date.  These are generally written in the form of comments. |  |  |  | | --- | --- | | Link | All header files are included in this section which contains different functions  from the libraries.  A copy of these header files is inserted into your code before compilation. |  |  |  | | --- | --- | | Definition | Includes preprocessor directive, which contains symbolic constants.  E.g.: #define allows us to use constants in our code. It replaces all the constants  with its value in the code. |  |  |  | | --- | --- | | Global Declaration | Includes declaration of global variables, function declarations, static global  variables, and functions. |  |  |  | | --- | --- | | Main() Function | For every C program, the execution starts from the main() function. It is mandatory to include a main() function in every C program. |  |  |  | | --- | --- | | Subprograms | Includes all user-defined functions (functions the user provides).  They can contain the inbuilt functions and the function definitions  declared in the Global Declaration section.  These are called in the main() function. | |
|  |  | C திட்டத்தின் அடிப்படை கட்டமைப்பை விளக்குங்கள். |
| **(OR)** | | |
|  | b) | Write the rules to form identifier/variable name.   * A variable can have alphabets, digits, and underscore. * A variable name can start with the alphabet, and underscore only. It can't start with a digit. * No whitespace is allowed within the variable name. * A variable name must not be any reserved word or keyword, e.g. int, goto, etc. |
|  |  | அடையாளங்காட்டி/மாறி பெயரை உருவாக்க விதிகளை எழுதவும். |
|  |  |  |
|  | a) | Write the output of the following statements in C.  printf(“%d”, 9678); printf(“%6d”, 9678);  printf(“%2d”, 9678); printf(“%-6d”, 9678);  printf(“%06d”, 9678);  9678  9678  009678  9678  9678 |
|  |  | பின்வரும் அறிக்கைகளின் வெளியீட்டை C இல் எழுதவும்.  printf ("%d", 9678); printf ("% 6d", 9678);  printf ("% 2d", 9678); printf("%-6d", 9678);  printf("%06d", 9678); |
| **(OR)** | | |
|  | b) | Explain Type conversion in expression with suitable examples. Implicit type conversion Implicit type conversion, also known as *coercion* or *type juggling*, is an automatic type conversion by the  compiler. Some [programming languages](https://en.wikipedia.org/wiki/Programming_language) allow compilers to provide coercion; others require it.  In a mixed-type expression, data of one or more [subtypes](https://en.wikipedia.org/wiki/Subtyping) can be [converted](https://en.wikipedia.org/wiki/Operators_in_C_and_C%2B%2B) to a supertype as needed at [runtime](https://en.wikipedia.org/wiki/Run_time_(program_lifecycle_phase)) so that the program will run correctly. For example, the following is legal [C language](https://en.wikipedia.org/wiki/C_(programming_language)) code:   * If one of the operands is long double, the other will be converted to long double   and the result will be long double   * else, if one of the operands is double,   the other win be converted to double and the result will be double   * else, if one of the operands is float, the other will be converted to float and the   result will be float   * else, if one of the operands is unsigned long int, the other will be converted to   unsigned long the other is unsigned int, then int and the result will be  unsigned long int   * else. If one of the operands in long int and the other is unsigned int   (a) if unsigned int can be converted to long int, the unsigned int operand will be converted  as such and the result will be long int  (b)else, both operands will be converted to unsigned  long int and the result will be unsigned long int   * Eelse, if one of the operands is long int, the other will be converted to long int and the result will   be long int   * else, one of the operands is unsigned int, the other will be converted to unsigned   result will be unsigned int Explicit type conversion Explicit type conversion, also called type casting, is a type conversion which is explicitly defined within a program (instead of being done automatically according to the rules of the language for implicit type conversion). It is defined by the user in the program. |
|  |  | பொருத்தமான எடுத்துக்காட்டுகளுடன் வெளிப்பாட்டின் வகை மாற்றத்தை விளக்குங்கள். |
|  | a) | Write a program to print the given number is Even or Odd by using If else statement.  Int n;  Scanf(“%d”,&n);  If(a%2==0)  Printf(“Even”);  Else  Printf(“ODD”); |
|  |  | If else என்ற கூற்றைப் பயன்படுத்தி கொடுக்கப்பட்ட எண்ணை இரட்டை அல்லது ஒற்றைப்படை என அச்சிட ஒரு நிரலை எழுதவும். |
| **(OR)** | | |
|  | b) | Explain FOR statement used in C with an example.  #include <stdio.h>  int main() {  int i;  for (i = 1; i < 11; i++)  {  printf("%d ", i);  }  return 0;  }   * i is initialized to 1. * The test expression i < 11 is evaluated. Since 1 less than 11 is true,   the body of for loop is executed. This will print the **1** (value of i) on the screen.   * The update statement ++i is executed. Now, the value of i will be 2. Again,   the test expression is evaluated to true, and the body of for loop is executed. This will print **2** (value of i) on the screen.   * Again, the update statement ++i is executed and the test expression i < 11    is evaluated. This process goes on until i becomes 11.   * When i becomes 11, i < 11 will be false, and the for loop   terminates. |
|  |  | C இல் பயன்படுத்தப்படும் FOR statement உதாரணத்துடன் விளக்கவும். |
|  | a) | Explain Multi-dimensional array with an example. |
|  |  | பல பரிமாண வரிசையை எடுத்துக்காட்டுடன் விளக்குங்கள். |
| **(OR)** | | |
|  | b) | Write a program to find a factorial of given number using recursion.  Int multile(int n){  If(n>=1)  Return n\*multile(n-1);  Else  Return 1;  } |
|  |  | மறுநிகழ்வைப் பயன்படுத்தி கொடுக்கப்பட்ட எண்ணின் Factorial கண்டறிய ஒரு நிரலை எழுதவும். |
|  | a) | Explain array of structures with an example.   * The most common use of structure in C programming is an array of structures. * To declare an array of structure, first the structure must be defined and then an array   variable of that type should be defined.   * For Example − struct book b[10]; //10 elements in an array of structures of type ‘book’   #include <stdio.h>  #include <string.h>  struct student{     int id;     float percentage;  };  int main(){     int i;     struct student record[2];     record[0].id=1;     record[0].percentage = 86.5;        record[1].id=2;     record[1].percentage = 90.5;  } |
|  |  | ஒரு எடுத்துக்காட்டுடன் array of structres விளக்குங்கள். |
| **(OR)** | | |
|  | b) | Explain fprintf and fscanf functions with an example.  Fprint  The fprintf() function is used to write set of characters into file. It sends formatted output to a stream.  **Syntax:**   1. **int** fprintf(**FILE** \*stream, **const** **char** \*format [, argument, ...])   **Example:**   1. #include <stdio.h> 2. main(){ 3. **FILE** \*fp; 4. fp = fopen("file.txt", "w"); 5. fprintf(fp, "Hello file by fprintf...\n"); 6. fclose(fp); 7. }     Fscanf  The fscanf() function is used to read set of characters from file. It reads a word from the file and returns EOF at the end of file.  **Syntax:**   1. **int** fscanf(**FILE** \*stream, **const** **char** \*format [, argument, ...])   **Example:**   1. #include <stdio.h> 2. main(){ 3. **FILE** \*fp; 4. **char** buff[255]; 5. fp = fopen("file.txt", "r"); 6. **while**(fscanf(fp, "%s", buff)!=EOF){ 7. printf("%s ", buff ); 8. } 9. fclose(fp); 10. } |
|  |  | fprintf மற்றும் fscanf செயல்பாடுகளை எடுத்துக்காட்டுடன் விளக்குங்கள். |

**Part – C Answer any Three Questions 3 X 10 Marks = 30 Marks**

|  |  |  |
| --- | --- | --- |
|  |  | Explain the five arithmetic operators with an example to each.  There are five arithmetic operators, +, -, \*, *I,*and %, which respectively represent the processes of addition, subtraction, multiplication, division, and modulus. The modulus operator (%) gives the remainder when one integer is divided by another integer. All of the five operators have been described with examples of codes in Table.  C Arithmetic Operators |
|  |  | ஐந்து arithmetic ஆபரேட்டர்களை ஒவ்வொன்றிற்கும் ஒரு எடுத்துக்காட்டுடன் விளக்கவும். |
|  |  |  |
|  |  | Explain any four string handling functions used in C with suitable examples.   |  |  | | --- | --- | | **Function** | **Description** | | strncmp() | It compares two strings only to n characters. | | strcat() | It concatenates two strings and returns the concatenated string. | | strncat() | It concatenates n characters of one string to another string. | | strcpy() | It copies one string into another. | |
|  |  | C இல் பயன்படுத்தப்படும் நான்கு string கையாளுதல் செயல்பாடுகளை பொருத்தமான எடுத்துக்காட்டுகளுடன் விளக்குங்கள். |
|  |  |  |
|  |  | Write a program to find the biggest of three numbers using nested if else statement.  include <stdio.h>  int main() {  int n1, n2, n3;  scanf("%d %d %d", &n1, &n2, &n3);  if (n1 >= n2 && n1 >= n3)  printf("%.d is the largest number.", n1);  if (n2 >= n1 && n2 >= n3)  printf("%d is the largest number.", n2)  if (n3 >= n1 && n3 >= n2)  printf("%d is the largest number.", n3);  re |
|  |  | Nested if else என்ற கூற்றைப் பயன்படுத்தி மூன்று எண்களில் பெரிய எண்களைக் கண்டறிய ஒரு நிரலை எழுதவும். |
|  |  |  |
|  |  | Write a program to add two given matrices.   1. #include < stdio.h > 2. int main() 3. { 4. int m, n, c, d, first[10][10], second[10][10], sum[10][10]; 5. printf("Enter the number of rows and columns of matrix\n"); 6. scanf("%d%d", & m, & n); 7. printf("Enter the elements of first matrix\n"); 8. **for** (c = 0; c < m; c++) 9. **for** (d = 0; d < n; d++) scanf("%d", & first[c][d]); 10. printf("Enter the elements of second matrix\n"); 11. **for** (c = 0; c < m; c++) 12. **for** (d = 0; d < n; d++) scanf("%d", & second[c][d]); 13. printf("Sum of entered matrices:-\n"); 14. **for** (c = 0; c < m; c++) 15. { 16. **for** (d = 0; d < n; d++) 17. { 18. sum[c][d] = first[c][d] + second[c][d]; 19. printf("%d\t", sum[c][d]); 20. } 21. printf("\n"); 22. } 23. **return** 0;   } |
|  |  | கொடுக்கப்பட்ட இரண்டு அணிகளைச் கூட்டுவதற்கு ஒரு நிரலை எழுதவும். |
|  |  |  |
|  |  | Explain how to access structure members and also explain the structure initialization with suitable example. |
|  |  | Structure உறுப்பினர்களை எவ்வாறு அணுகுவது என்பதை விளக்கவும் மற்றும் Structure துவக்கத்தை பொருத்தமான உதாரணத்துடன் விளக்கவும். |

\*\*\*\*\*\*\*\*\*\*\*\*\*