Ct2- key - Ct2- key

Programming For Problem Solving (SRM Institute of Science and Technology)

SRM INSTITUTE OF SCIENCE & TECHNOLOGY (Deemed to be University u/s 3 of UGC Act, 1956)	**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY** **Vadapalani, Chennai – 600 026**	**Register No:**

CONTINOUS LEARNING (THEORY) ASSESSMENT– II
ODD SEMESTER, 2020- 2021
B.Tech COMPUTER SCIENCE AND ENGINEERING
18CSS101J – PROGRAMMING FOR PROBLEM SOLVING

Date: 12:1:2021 Time: 90 Mins SEMESTER-I MAXIMUM MARKS:50

COURSE LEARNING RATIONALE:

CLR -1: Think and evolve a logically to construct an algorithm into a flowchart and a pseudo code that can be programmed
CLR -2: Utilize the logical operators and expressions to solve problems in engineering and real time
CLR -3: Store and retrieve data in a single and multidimensional array
CLR -4: Utilize custom designed functions that can be used to perform tasks and can be repeatedly used in any application
CLR -5: Create storage constructs using structure and union, create and utilize files to store and retrieve information
CLR -6: Create a logical mindset to solve various engineering applications using programming constructs in C

COURSE LEARNING OUTCOME:

CLO-1: Identify methods to solve a problem through computer programming. List the basic data types and variables in C.
CLO-2 : Apply the logic operators and expressions. Use loop constructs and recursion. Use array to store and retrieve data.
CLO-3 : Analyze programs that need storage and form single and multi-dimensional arrays. Use preprocessor constructs in C.

	PART – A **Answer the MCQ's (10*1=10 Marks)**	**CLO.**	**K.NO.**
1.	Which of the following operator takes only integer operands? A) + B) / C) % D) None of these **ANSWER:C**	CLO1	K1
2.	In an expression involving || operator, evaluation I) Will be stopped if one of its components evaluates to false II) Will be stopped if one of its components evaluates to true III) Takes place from right to left IV) Takes place from left to right A) I and II B) I and III C) II and III	CLO2	K2

	D) II and IV **ANSWER:D**		

3.	Find the output of the following:
	void main()
	{
	int i=0, j=1, k=2, m;
	m= i++ || j++ || k++ ;
	printf("%d %d %d %d", m, i, j, k);
	}
	
	A) 1 1 2 3
	B) 1 1 2 2
	C) 0 1 2 2
	D) 0 1 2 3
	ANSWER:B
		CLO2	K2

4.	What will be the output?
	void main()
	{
	int a=10, b=20;
	char x=1, y=0;
	if(a,b,x,y)
	printf("EXAM");
	}
	
	A) XAM is printed
	B) EXAM is printed
	C) COMPILER ERROR
	D) Nothing is printed
	ANSWER:D
		CLO2	K2

5.	Which operator from the following has the lowest priority? A) Assignment B) Division C) Comma D) Dot **ANSWER:C**	CLO2	K1
6.	Array elements are always stored in --------- memory locations. A) Sequential B) Random C) Sequential and Random D) None of these **ANSWER:A**	CLO3	K1
7.	Let x be an array. Which of the following operations are illegal (throws error)? I) ++x II) x+1 III) x++ IV) x*2 A) I and II B) III and IV	CLO3	K1

	C) I, II and III D) I, III and IV **ANSWER:B**		
8.	What is the output of C Program.? int main() { int a[3] = {10,12,14}; a[1]=20; int i=0; while(i<3) { printf("%d ", a[i]); i++; } } A) 20 12 14 B) 10 20 14 C) 10 12 20 D) Compiler error **ANSWER:B**	CLO3	K2
9.	What is the Format specifier used to print a String or Character array in C Printf or Scanf function? A) %c B) %C C) %s D) %w **ANSWER:C**	CLO3	K1

10.	What is the output of this C code?
	
	int main()
	{
	void foo(), f();
	f();
	}
	void foo()
	{
	printf("2 ");
	}
	void f()
	{
	printf("1 ");
	foo();
	}
	
	A. Compile time error as foo is local to main
	B. 1 2
	C. 2 1
	D. Compile time error due to declaration of functions inside main
	ANSWER:B
		CLO3	K2

PART – B **Answer ANY FOUR questions(4 x 4 = 16 Marks)**			
11.	Write C program to input a sequence of characters from user and check whether given character array has an alphabet or digit or special character using if else. **ANSWER:** **Logic:** A character is alphabet if it in between a-z or A-Z.	CLO2	K2

	A character is digit if it is in between 0-9.
	A character is special symbol character if it neither alphabet nor digit.
	
	Step by step descriptive logic to check alphabet, digit or special character.
	
	1. Input a character from user. Store it in some variable say ch.
	2. First check if character is alphabet or not. A character is alphabet if((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z')).
	3. Next, check condition for digits. A character is digit if(ch >= '0' && ch <= '9').
	4. Finally, if a character is neither alphabet nor digit, then character is a special character.
	
	Implement the above logic in a C program as follows:
	
	`#include <stdio.h>`
	
	`int main()`
	`{`
	`char ch;`
	
	`/* Input character from user */`
	`printf("Enter any character: ");`
	`scanf("%c", &ch);`
	
	`/* Alphabet check */`
	`if((ch >= 'a' && ch <= 'z') || (ch >= 'A' && ch <= 'Z'))`
	`{`
	`printf("'%c' is alphabet.", ch);`
	`}`
	`else if(ch >= '0' && ch <= '9')`
	`{`
	`printf("'%c' is digit.", ch);`
	`}`
	`else`
	`{`
	`printf("'%c' is special character.", ch);`
	`}`
	
	`return 0;`
	`}`
	SAMPLE INPUT and OUTPUT:
	`Enter any character: a`
	`'a' is alphabet.`		

12.	Discuss about the Conditional iterators in C language. **ANSWER:** In computer programming, conditional loops or repetitive control structures are known as conditional iterators. **while loop syntax:** ***<initialization>*** **while (*<test>*)** **{**	CLO2	K1

	<*statement1*>;
	...
	<*statementN*>;
	<*update*>
	}
	for loop syntax:
	for (<*initialization*>;
	<*test*>;
	<*update*>)
	{
	<*statement1*>;
	...
	<*statementN*>;
	}
	Example:
	Calculate the sum of the integer numbers between 1 and 10
	int sum = 0; // this program piece
	int i = 1; // calculates the sum of
	while (i <= 10) // integers between and
	{ // including 1 and 10
	sum = sum + i;
	i = i + 1;
	}
	printf("Sum=%d",sum);
	Same example with for loop
	int sum = 0;
	for (int i=1; i <= 10; i=i+1)
	{
	sum = sum + i;
	}
	printf("Sum=%d",sum);
	
	SAMPLE INPUT and OUTPUT:
	Sum = 55		

13.	Write short notes on: strrev, strcpy, strstr, strtok
	ANSWER:
	STRREV():
	● reverses a given string in C language.
	Syntax:
	char *strrev(char *string);
	Example :
	char name[30] = "Hello";
	printf("String before strrev() :%s\n",name);
	printf("String after strrev(%s", strrev(name));
	Output:
	String before strrev() : Hello
	String after strrev() : olleH
	STRCPY FUNCTION:
	Copy second string into First
		CLO3	K1

	Syntax :
	char * strcpy (char *string1, char *string2) ;
	Example :
	char s1[10] = "SAM" ;
	char s2[10] = "MIKE" ;
	strcpy (s1,s2);
	puts (s1) ; // Prints : MIKE
	puts (s2) ; // Prints : MIKE
	Output:
	MIKE
	MIKE
	STRSTR FUNCTION:
	● Finds first occurrence of sub-string in other string
	Syntax:
	char *strstr(const char *s1, const char *s2);
	Features :
	● On success, strstr returns a pointer to the element in s1 where s2 begins (points to s2 in s1).
	● On error (if s2 does not occur in s1), strstr returns null.
	Example :
	char string[55] ="This is a test string;
	char *p;
	p = strstr (string,"test");
	if(p){
	printf("string found\n"); }
	else
	printf("string not found\n");
	Output : string found
	STRTOK FUNCTION
	● **tokenizes/parses the given string using delimiter.**
	Syntax
	char * strtok (char * str, const char * delimiters);
	Example :
	char str[] = "Problem_Solving_in_c";//Returns first token
	char* token = strtok(str, "_");//Keep printing tokens while one of the delimiters present in str[].
	while (token != NULL) {
	printf("%s\n", token);
	token = strtok(NULL, "_");
	}
	Output:
	Problem
	Solving
	in
	C		

14.	How an array construction is done for real-time application and what are the common programming errors in it? **ANSWER:**	CLO3	K1

(i) Constant Expression Require

```
#include<stdio.h>
void main()
{
int i=10;
int a[i];
}
```

In this example we see what's that error?

- We are going to declare an array whose size is equal to the value of variable.
- If we changed the value of variable then array size is going to change.
- According to array concept, we are allocating memory for array at compile time so if the size of array is going to vary then how it is possible to allocate memory to an array.
- *i is initialized to 10 and using a[i] does not mean a[10]because 'i' is Integer Variable whose value can be changed inside program.*

– **Value of Const Variable Cannot be changed**

– we know that value of Const Variable cannot be changed once initialized so we can write above example as below –

```
#include<stdio.h>
void main()
{
const int i=10;
int a[i];
}
```

or

int a[10];

(ii) Empty Valued 1D Array

```
#include<stdio.h>
void main()
{
int arr[];
}
```

Instead of it Write it as –

```
#include<stdio.h>
void main()
{
int a[] = {1,1};
}
```

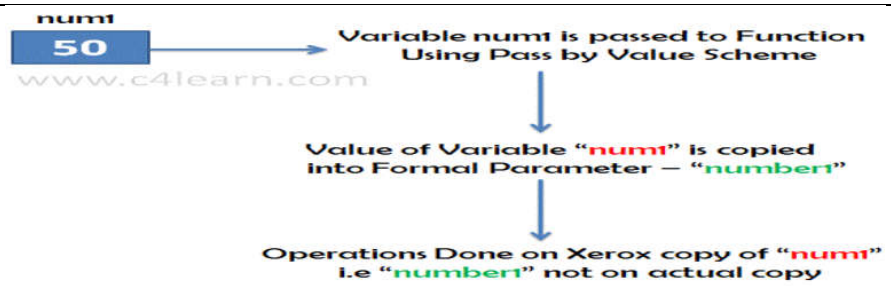
• Consider this example, we can see the empty pair of square brackets means we haven't specified size of an 1D Array. In this example array 'arr' is undefined or empty.

• Size of 1D Array should be Specified as a Constant Value.

```
#include<stdio.h>
void main()
{
int a[] = {};// This also Cause an Error
}
```

	(iii) 1D Array with no Bound Checking
	#include<stdio.h>
	void main()
	{
	int a[5];
	printf("%d",a[7]);
	}
	Here Array size specified is 5.
	● So we have Access to Following Array Elements – a[0],a[1],a[2],a[3] and a[4].
	● But accessing a[5] causes Garbage Value to be used because C Does not performs Array Bound Check.
	If the maximum size of array is "MAX" then we can access following elements of an array –
	Elements accessible for Array Size "MAX" = arr[0]= arr[MAX-1]
	
	(iv) Case Sensitive
	#include<stdio.h>
	void main()
	{
	int a[5];
	printf("%d",A[2]);
	}
	Array Variable is Case Sensitive so A[2] does not print anything it Displays Error Message : "Undefined Symbol A"		

15.	Compare the call by value and reference in function call with an example.
	ANSWER:
	
	Call by value This method copies the actual value of an argument into the formal parameter of the function. In this case, changes made to the parameter inside the function have no effect on the argument.
	/* function definition to swap the values */
	void swap (int x, int y)
	{
	int temp; temp = x;
	/* save the value of x */
	x = y; /* put y into x */
	y = temp; /* put temp into y */
	return;
	}
		CLO3	K2



Call by reference This method copies the address of an argument into the formal parameter. Inside the function, the address is used to access the actual argument used in the call. This means that changes made to the parameter affect the argument.

```
/* function definition to swap the values */
void swap (int x, int y)
{
int temp;
temp = *x;
*x = *y;
*y = temp;
return; }
```

PART – C

Answer the following **(2 x 12 = 24 Marks)**

16. (a) Enumerate and compare **for** loop syntax with while loop using an example in C language. Discuss the common problems in loops. — CLO2 — K2

ANSWER:

<u>while loop syntax:</u>

```
<initialization>
while (<test>)
{
        <statement1>;
        ...
        <statementN>;
  <update>
}
```

<u>for loop syntax:</u>

```
for (<initialization>;
        <test>;
        <update> )
{
        <statement1>;
    ...
  <statementN>; }
```

Comparison

for loop syntax compared with while

```
<initialization>
while (<test>)
{
      <statement1>;
      ...
      <statementN>;
      <update>
}
```

```
for (<initialization>;
      <test>;
      <update> )
{
      <statement1>;
      ...
      <statementN>;
}
```

Example:

Calculate the sum of the integer numbers between 1 and 10

```
int sum = 0;          // this program piece
int i = 1;            // calculates the sum of
while (i <= 10)       // integers between and
{                           // including 1 and 10
  sum = sum + i;
  i = i + 1;
}
printf("Sum=%d",sum);
```

Same example with for loop

```
int sum = 0;
for (int i=1; i <= 10; i=i+1)
{
        sum = sum + i;
}
printf("Sum=%d",sum);
```

SAMPLE INPUT and OUTPUT:

Sum = 55

(OR)

(b)	Write a C program to find maximum and minimum between two numbers using functions **ANSWER:** Embed the logic to find maximum within a function. So define function to find maximum. 1. First give a meaningful name to our function. Say max() function is used to find maximum between two numbers.	CLO3	K2

2. Second, we need to find maximum between two numbers. Hence, the function must accept two parameters of int type say, max(int num1, int num2).
3. Finally, the function should return maximum among given two numbers. Hence, the return type of the function must be same as parameters type i.e. int in our case.

After combining the above three points, function declaration to find maximum is int max(int num1, int num2);.

```
#include <stdio.h>

/* Function declarations */
int max(int num1, int num2);
int min(int num1, int num2);



int main()
{
    int num1, num2, maximum, minimum;

    /* Input two numbers from user */
    printf("Enter any two numbers: ");
    scanf("%d%d", &num1, &num2);

    maximum = max(num1, num2);  // Call maximum function
    minimum = min(num1, num2);  // Call minimum function

    printf("\nMaximum = %d\n", maximum);
    printf("Minimum = %d", minimum);

    return 0;
}

/**
 * Find maximum between two numbers.
 */
int max(int num1, int num2)
{
    return (num1 > num2 ) ? num1 : num2;
}

/**
 * Find minimum between two numbers.
 */
int min(int num1, int num2)
{
    return (num1 > num2 ) ? num2 : num1;
}
```

SAMPLE INPUT and OUTPUT:

```
Enter any two numbers: 10 20
```

	Maximum = 20 Minimum = 10		
17. (a)	Write a C program to compute addition of two matrices. **ANSWER:**

```c
#include <stdio.h>
int main() {
    int r, c, a[100][100], b[100][100], sum[100][100], i,
j;
    printf("Enter the number of rows (between 1 and 100):
");
    scanf("%d", &r);
    printf("Enter the number of columns (between 1 and
100): ");
    scanf("%d", &c);

    printf("\nEnter elements of 1st matrix:\n");
    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j) {
            printf("Enter element a%d%d: ", i + 1, j +
1);
            scanf("%d", &a[i][j]);
        }

    printf("Enter elements of 2nd matrix:\n");
    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j) {
            printf("Enter element a%d%d: ", i + 1, j +
1);
            scanf("%d", &b[i][j]);
        }

    // adding two matrices
    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j) {
            sum[i][j] = a[i][j] + b[i][j];
        }

    // printing the result
    printf("\nSum of two matrices: \n");
    for (i = 0; i < r; ++i)
        for (j = 0; j < c; ++j) {
            printf("%d   ", sum[i][j]);
            if (j == c - 1) {
                printf("\n\n");
            }
```
 | CLO2 | K3 |

	} return 0; } **SAMPLE INPUT and OUTPUT:** Enter the number of rows (between 1 and 100): 2 Enter the number of columns (between 1 and 100): 3 Enter elements of 1st matrix: Enter element a11: 2 Enter element a12: 3 Enter element a13: 4 Enter element a21: 5 Enter element a22: 2 Enter element a23: 3 Enter elements of 2nd matrix: Enter element a11: -4 Enter element a12: 5 Enter element a13: 3 Enter element a21: 5 Enter element a22: 6 Enter element a23: 3 Sum of two matrices: -2 8 7 10 8 6		
	(OR)		
(b)	Write a C program to convert given uppercase string to lowercase and viceversa. **ANSWER:** String is a sequence of characters. char data type is used to represent one single character in C. So if you want to use a string in your program then you can use an array of characters. The declaration and definition of the string using an array of chars is similar to declaration and definition of an array of any other data type. Any string ends with a terminating null character '\0'. An array definition in such a way should include null character '\0' as the last element. #include<stdio.h> #include<string.h> int main() { char str[20];	CLO3	K3

```
        int i;
        printf("\nEnter any string :: ");
        gets(str);
         printf("\nThe input String is :: [ %s ]\n",str);
        for(i=0;i<=strlen(str);i++)
    {
            if(str[i]>=97&&str[i]<=122)
                str[i]=str[i]-32;
            else if(str[i]>=65&&str[i]<=90)
                str[i]=str[i]+32;
            else;
        }
        printf("\nThe Converted String is :: [ %s ]\n",str);
         return 0;
}
```

SAMPLE INPUT and OUTPUT:

Enter any string :: CodezClub
The input String is :: [CodezClub]
The Converted String is :: [cODEZcLUB]

K1- Remember, K2 - Understand, K3 – Apply, K4- Analyse, K5 – Evaluate, K6 - Create