Set C- Answer Key CT-3 - PPS CLAT - 3 SET-C QUESTION PAPER

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

SRM Institute of Science and Technology
Faculty of Engineering and Technology
School of Computing

Set C

DEPARTMENT OF COMPUTING TECHNOLOGIES

SRM Nagar, Kattankulathur – 603203, Chengalpattu District, Tamilnadu

Academic Year: 2021 – 2022 EVEN

Test: CLAT-3 **Date:**
Course Code & Title: 18CSS101J & Programming for Problem Solving **Duration:** 2 periods
Year & Sem: I & II **Max. Marks: 50**

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	3	-	-	-	-	-	-	-	-	-	-	-	-	3
CO2	3	3	2	-	-	-	-	-	-	-	-	-	-	-	3
CO3	3	3	3	-	-	-	-	-	-	-	-	-	-	-	3
CO4	3	2	3	3	-	-	-	-	-	-	-	-	-	-	3
CO5	3	3	3	-	-	-	-	-	-	-	-	2	-	-	3
CO6	2	2	-	-	-	-	-	-	-	-	-	2	-	-	3

Part – A
(10 x 1 = 10 Marks)
Instruction: Answer all the questions

Q. No	Answer with choice variable	Marks	BL	CO	PO	PI Code
1	Which of the following is the correct syntax to send an array as a parameter to function? **Answer: a) func(array);**	1	1	3	1	1.7.1
2	The keyword used to define a C macro is ------- **Answer:c) define**	1	1	3	1	1.7.1
3	The symbol * and &amp represents _____ respectively in pointers. **Answer:c) Value of operator and Address at operator**	1	1	4	1	1.7.1
4	Identify the output of the following program #include <stdio.h> int *money() { int *pocket = 5; return pocket; } void main() { int *kit = money(); printf("%d", kit); } **Answer:a) 5**	1	2	4	1	1.7.1
5	What is the output of this C code?	1	2	4	1	1.7.1

	`` #include <stdio.h> void foo(int*); int main() { int i = 10; foo((&i)++); } void foo(int *p) { printf("%d\n", *p); } ``` **Answer: Compile time error is the correct answer** a) 10 b) 9 c) 11 d) 12					
6	How structure, union, and enumeration are connected? a. By defining new data types b. By defining new variable c. By defining new structures d. By defining new pointers **Answer:- a**	**1**	**2**	**5**	**1**	**1.7.1**
7	Predict the value of 'm' in the following code #include <stdio.h> int main() { unsigned int s = 5; unsigned int r = 7; int m = 0; m = s ^ r; printf("%d\n", m); return 0; } a) 1 b) 2 c) 3 d) 4 **Answer:- b**	**1**	**2**	**5**	**1**	**1.7.1**
8	Size of the union defined in the following code is ---- #include<stdio.h> union srm { int x; float y; double z; }; int main() { printf("%lu", sizeof(union srm)); return 0; } a) 2 b) 4 c) 8 d) 16 **Answer:- c**	**1**	**2**	**5**	**1**	**1.7.1**
9	Dynamic Memory Allocation means a) memory can be allocated at compile time. b) faster than that using static memory allocation. c) memory can't be increased while executing program. d) memory can be allocated at run time. **Answer:- d**	**1**	**1**	**5**	**1**	**1.7.1**
10	The equivalent memory function(s) which has the same effect as calloc ----------	**1**	**1**	**5**	**1**	**1.7.1**

	a) malloc and memset b) realloc c) malloc d) recalloc and memset **Answer:- a**					
Part – B **(4 x5 = 20 Marks)** **Instruction : Answer all the questions**						
11	Write a macro to determine whether the given number is odd or even **Answer:** #include<stdio.h> #define ODD_EVEN(num) ((num % 2 == 0) ? printf("Even\n") : printf("ODD\n")) int main() { int num; printf("Enter a positive number\n"); scanf("%d", &num); ODD_EVEN(num); return 0; }	**5**	**3**	**3**	**2**	**2.5.2**
12	The marks scored by Stella in 6 subjects are given and Bala was asked to find out the total marks of Stella. Help Bala to write a program for this using array of pointers. **Answer :** Marks scored can be maintained in an array and summed using pointers as below `#include<stdio.h>` `int main()` `{` `int mark[6]={45,56,67,78,89,91},sum=0;` `int i;` `int *p[6];` `for (i=0;i<6;i++)` `{` `p[i]=&mark[i];` `}` `for (i=0;i<6;i++)` `{` `sum = sum + *p[i];` `}` `printf("Sum= d ", sum);` `return 0;` `}`	**5**	**4**	**4**	**2**	**2.5.2**
13	Mention various library functions and its purpose of dynamic memory allocation in C. **Answer**: i. malloc() - Used to allocate a **single block of memory** to store values of specific data types. ii. calloc() - used to allocate **memory in multiple blocks of same size** during program execution. iii. free() - memory at **location pointed to by pointer is released.** iv. realloc() - used to modify or **reallocate the memory space which is previously allocated**.	**5**	**3**	**5**	**2**	**2.5.2**

14	Our SRM Institute is conducting an inter collegiate Competition named “Problem Solving through Programming in C”. In that competition, suppose your class student scored a centum mark and got First Prize. Now it is your duty to write a c program to display the concerned student register number, name, contest _name, and marks scored. **Answer:-** **Using typedef struct** #include <stdio.h> #include <string.h> typedef struct Students { char reg_no[15]; char name[50]; char contest_name [100]; int marks; char prize[15]; } Student; int main( ) { Student stud; strcpy(stud.reg_no, "100479"); strcpy(stud.name, "Manikandan K"); strcpy(stud.contest_name, " Problem Solving through Programming in C "); stud.marks = 100; strcpy(stud.prize, "FIRST"); printf("Student Reg_no : %s\n", stud.reg_no); printf("Student Name : %s\n", stud.name); printf("Student contest_name: %s\n", stud. contest_name); printf("Student Marks : %d\n", stud.marks); printf("Student Prize : %s\n", stud. prize); return 0; } **Output:** Student Reg_no : 100479 Student Name : Manikandan K Student contest_name: Problem Solving through Programming in C Student Marks : 100 Student Prize : FIRST	**5**	**4**	**5**	**2**	**2.5.2**
Part – c **(2x10 = 20 Marks)** **Instruction : Answer all the questions**						
15 a	In a certain code language, numerals are each represented by a symbol or a letter. They are as follows	**10**	**4**	**3**	**2**	**2.5.2**

Numeral	0	1	2	3	4	5	6	7	8	9
Symbol code	*	B	E	A	@	F	K	%	R	M

Numerals are to be coded as per the codes in the table and the following conditions:

a. If the first and the last digits are odd, both are to be coded as 'X'.

b. If the first and the last digits are even, both are to be coded as '$'.

c. If the last digit is '0', it is to be coded as '#'.

For example, 487692 is coded as $R%KM$. Write a program with a function that takes as input a sequence of numbers in an array and displays the corresponding code to represent it.

Answer:

```
#include<stdio.h>
char *symbol_code="*BEA@FK%RM";
int a[20],n,i;
char final_code[30];
int main()
{
printf("Enter the no of digits of input number");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("Enter digit %d",i+1);
scanf("%d",&a[i]);
final_code[i]=symbol_code[a[i]];
}
if(a[0]%2==1 && a[n-1]%2 ==1)
{
final_code[0]='X';
final_code[n-1]='X';
}
if(a[0]%2==0 && a[n-1]%2 ==0)
{
final_code[0]='$';
final_code[n-1]='$';
}
if(a[n-1]==0)
{
final_code[n-1]='#';
}
printf("Final code is %s",final_code);
}
```

OR

b Ravi has given positive integers to Kumar and asked him to sort the list. Kumar wants to solve the problem with the help of pointers. Can you help him in writing the solution? **10** **4** **4** **2** **2.5.2**

Answer:

```
#include <stdio.h>
void main()
{
int *a,i,j,tmp,n;
printf("Enter the number of elements to store in the array: ");
scanf("%d",&n);
printf("Enter the elements");
  for(i=0;i<n;i++)
    {
          printf(" element - %d : ",i+1);
          scanf("%d",a+i);
          }
  for(i=0;i<n;i++)
```

	{ for(j=i+1;j<n;j++) { if(*(a+i) > *(a+j)) { tmp = *(a+i); *(a+i) = *(a+j); *(a+j) = tmp; } } } printf("\n The elements in the array after sorting : \n"); for(i=0;i<n;i++) { printf(" element - %d : %d \n",i+1,*(a+i)); } printf("\n"); }					
16 a	Write a program to multiply two integers with bitwise operators using functions in C. **Answer:-** **Russian Multiplication:** The powers of two in the decomposition of the multiplicand are found by writing it on the left and progressively halving the left column, discarding any remainder, until the value is 1 (or −1, in which case the eventual sum is negated), while doubling the right column as before. Lines with even numbers on the left column are struck out, and the remaining numbers on the right are added together. #include <stdio.h> int add (int x, int y) { while (y!= 0) { int carry = x & y; x = x ^ y; y = carry << 1; } return x; } int main () { int a, b, result; printf ("\nEnter the numbers to be multiplied:") ; scanf ("%d%d", &a, &b); result = 0; while (b!= 0) { if (b & 1)	**10**	**3**	**5**	**2**	**2.5.2**

	result = add (result, a); a <<= 1; b >>= 1; } printf ("\nResult: %d", result); return 0; } **Output**:- Enter the numbers to be multiplied: 5 7 Result : 35					
	OR					
b	Discuss about File Handling in C with syntax and suitable example **Answer:- Definition & Operation Mode – 2 Mark** **Any Four Operations with syntax and example – 4 * 2 = 8 Marks** **Definition**: A file is a container in computer storage devices which is used for storing output or information permanently in the form of a sequence of bytes on the disk. **Operations on File: -** 1. fopen() opens new or existing file 1. fprintf() write data into the file 2. fscanf() reads data from the file 3. fputc() writes a character into the file 4. fgetc() reads a character from file 5. fclose() close the file 6. fseek() sets the file pointer to given position 7. fputw() reads an integer from file 8. getw() reads an integer from file 9. ftell() returns current position 10. Rewind() sets the file pointer to the beginning of the file **File Opening Modes** r – Open text file read mode. If file exists, the marker is positioned beginning. If file doesn't exist, it is created. w- Open text file in write mode • If file exists, it is erased.	**10**	**3**	**5**	**2**	**2.5.2**

- If file exists, the marker is positioned beginning

a- Open text file in append mode

- If file exists, the marker is positioned at end.
- If file doesn't exist, it is created.

Additional Modes

- r+ - open for reading and writing, start at beginning
- w+ - open for reading and writing (overwrite file)
- a+ - open for reading and writing (append if file exists)

File Open

Syntax:

```
FILE *fp;

fp=fopen ("filename", "'mode");
```

Example:

spData = fopen("MYFILE.TXT", "w");

spData = fopen("A:\\MYFILE.TXT", "w");

Closing a File

Syntax:

int **fclose**(FILE *fp);

Example:

fclose (fp);

fscanf:-

Syntax:

fscanf (fp,"string",identifiers);

Example:

FILE *fp;

Fp=fopen("input.txt","r"); int I;

fscanf (fp,"%d",i);

fprintf:-

Syntax:

fprintf (fp,"string",variables);

Example:

```
int I = 12;
float x = 2.356;  char ch = 's';  FILE *fp;
fp=fopen("out.txt","w");
fprintf (fp, "%d %f %c", I, x, ch);
```

getc()

Syntax:

identifier = getc (file pointer);

Example:

```
FILE *fp;
fp=fopen("input.txt","r");  char ch;
ch = getc (fp);
```

putc()

write a single character to the output file, pointed to by fp.

Example:

```
FILE *fp;  char ch;
putc (ch,fp);
```

End of File

- There are a number of ways to test for the end-of-file condition. Another way is to use the value returned by the *fscanf* function:

```
FILE *fptr1;
int istatus ;
istatus = fscanf (fptr1, "%d", &var) ;
if ( istatus == feof(fptr1) )
{
printf ("End-of-file encountered.\n") ;
}
```

Reading and Writing Files

```
#include <stdio.h>
int main ( )
```

```
{
        FILE *outfile, *infile ;
        int b = 5, f ;
        float a = 13.72,
        c = 6.68, e, g ;
        outfile = fopen (“testdata”, “w”) ;
        fprintf (outfile, “ %f %d %f “, a, b, c) ;
        fclose (outfile) ;
        infile = fopen (“testdata”, “r”) ;
        fscanf (infile,”%f %d %f”, &e, &f, &g) ;
        printf (“ %f %d %f \n “, a, b, c) ;
        printf (“ %f %d %f \n “, e, f, g) ;
}
```

fread **()**

Declaration:

```
size_t fread(void *ptr, size_t size, size_t n, FILE
        *stream);
```

Remarks:

fread reads a specified number of equal-sized data items from an input stream into a block.

Ptr = Points to a block into which data is read

size = Length of each item read, in bytes

n = Number of items read stream = file pointer

Example:

```
#include <stdio.h>
int main()
{
        FILE *f;
        char buffer[11];
        if (f = fopen(“fred.txt”, “r”))
        {
                fread(buffer, 1, 10, f);
                buffer[10] = 0;  fclose(f);
                printf(“first 10 characters of the file:\n
                                %s\n”, buffer);
        }
return 0;
```

fwrite()

Declaration:

```
 size_t fwrite(const void *ptr, size_t size, size_t
        n, FILE*stream);
```

Remarks:

fwrite appends a specified number of equal-sized data items to an output file.

Ptr = Pointer to any object; the data written begins at ptr

size = Length of each item of data

n =Number of data items to be appended

stream = file pointer

Example:

```
#include <stdio.h>
int main()
{
   char a[10]={'1','2','3','4','5','6','7','8','9','a'};
   FILE *fs;
   fs=fopen("Project.txt","w");
   fwrite(a,1,10,fs);
   fclose(fs);
   return 0;
}
```

fseek()

This function sets the file position indicator for the stream pointed to by stream or you can say it seeks a specified place within a file and modify it.

SEEK_SET - Seeks from beginning of file

SEEK_CUR - Seeks from current position

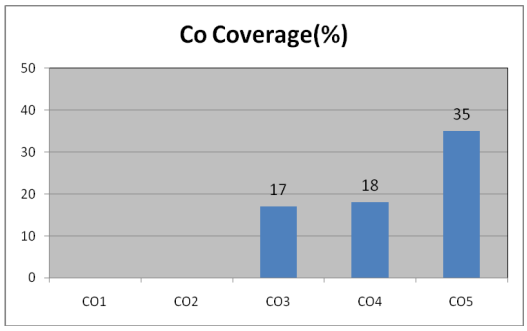
SEEK_END – Seeks from end of file

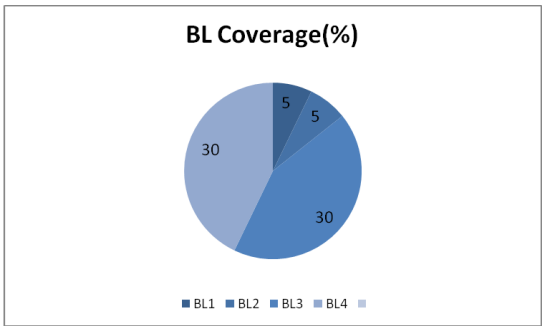
Example:

```
#include <stdio.h>
int main(){
      FILE * f;
      f = fopen("myfile.txt", "w");
      fputs("Hello World", f);
      fseek(f, 6, SEEK_SET);          SEEK_CUR,
            SEEK_END
      fputs(" India", f);
      fclose(f);
      return 0;}
```

***Performance Indicators are available separately for Computer Science and Engineering in AICTE examination reforms policy.**

Course Outcome (CO) and Bloom's level (BL) Coverage in Questions





Approved by the Audit Professor/Course Coordinator