Set B -answer key CT-3 - PPS CLAT - 3 SET-B 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 B

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	In C, if you pass an array as an argument to a function, what actually gets passed? **Answer: c) Base address of the array**	1	1	3	1	1.7.1
2	At which stage the following code **#include<stdio.h>** gets replaced by the contents of the file stdio.h **Answer: a) During Preprocessing**	1	1	3	1	1.7.1
3	How many numbers of pointer (*) does C have against a pointer variable declaration? **Answer: d) No limits.**	1	1	4	1	1.7.1
4	Predict the output of the given code snippet `#include <stdio.h>` `void foo(int*);` `int main()` `{` `  int i = 10, *p = &i;` `  foo(p++);` `}` `void foo(int *p)` `{` `  printf("%d\n", *p);` `}` **Answer: a) 10**	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 main() { int k = 5; int *p = &k; int **m = &p; **m = 6; printf("%d\n", k); } ``` **Answer :c) 6**					
6	Identify the output of the following program int main() { struct ship { int size; char color[10]; }boat1, boat2; boat1.size=10; boat2 = boat1; printf("boat2=%d",boat2.size); return 0; } a) boat2=0 b) boat2=-1 **c) boat2=10** d) Compiler error Answer: C	**1**	**2**	**5**	**1**	**1.7.1**
7	The size of the following union declaration is----- (Assuming size of double = 8, size of int = 4, size of char = 1) #include <stdio.h> union uTemp { double a; int b[10]; char c; }u; a)4 b) 8 **c) 40** d) 80 Answer: C	1	**2**	**5**	**1**	**1.7.1**
8	What will be the output of the following C code? #include <stdio.h> typedef int integer; int main() { int i = 10, *ptr; float f = 20; integer j = i; ptr = &j; printf("%d\n", *ptr); return 0; } a) 200 b) 100 c) 20 **d) 10** Answer: D	**1**	**2**	**5**	**1**	**1.7.1**
9	Choose a correct syntax for FSCANF and FPRINTF in c language. a) fprintf("format specifier",variables, fp); fscanf("format specifier",variables, fp); b) fprintf(fp,count,"format specifier",variables); fscanf(fp,count,"format specifier",variables); **c) fprintf(fp,"format specifier",variables); fscanf(fp,"format specifier",variables);** d) fprintf(fp,"format specifier",variables); fscanf("format specifier",variables); Answer: C	**1**	**1**	**5**	**1**	**1.7.1**
10	What is the output of this program? #include<stdio.h> #include<stdlib.h> int main() { int *ptr1, *ptr2; ptr1 = malloc(4); *ptr1 = 10;	**1**	**1**	**5**	**1**	**1.7.1**

	*ptr2 = free(ptr1); printf("%d\n",*ptr2); return 0; } a) 10 b) it will print the address stored in ptr1 c) it will print the address stored in ptr2 d) **it will give an error** Answer: D					
Part – B (4 x5 = 20 Marks) Instruction : Answer all the questions						
11	Define a macro to find the maximum of three integers **Answer:** #include<stdio.h> #define max(x, y, z) ((x > y && x > z) ? x : (y > z) ? y : z) int main() { int a, b, c; printf("Enter 3 integer numbers\n"); scanf("%d%d%d", &a, &b, &c); printf("Maximum of 3 numbers is %d\n", max(a, b, c)); return 0; }	**5**	**3**	**3**	**2**	**2.5.2**
12	A class teacher wants to maintain the list of top 5 students' names. She tries to develop a program for this using array of pointers. Is it possible? Justify your answer. **Answer:** Yes.It is possible to maintain the list of top 5 students' names using an array of pointers **Justification:** An array of pointers to strings is an array of character pointers where each pointer points to the first character of the string or the base address of the string. **Declaration and initialization of an array of pointers to strings:** char *topper[5] = { "Michal", "Celine", "Tina", "Robert", "James" }; Each element of the 'topper' array is a string literal and since a string literal points to the base address of the first character, the base type of each element of the 'topper' array is a pointer to char or (char*). The 0th element i.e topper[0] points to the base address of string "Michal". Similarly, the 1st element i.e topper[1] points to the base address of string "Celine" and so on. toppers 1st →Michal 2nd →Celine 3rd →Tina 4th →Robert 5th →James	**5**	**4**	**4**	**2**	**2.5.2**
13	Tavisha is getting the student's information such as roll_no, fname and lname of 5 students and she wanted to prints the records in ascending order on the basis of roll_no. she is quite confused to accomplish this task. Can you please help her to do the same? for(i=0; i<5; i++) { printf("\n Enter roll number:"); scanf("%d", &s[i].roll_no); printf("\n Enter first name:");	**5**	**4**	**5**	**2**	**2.5.2**

	scanf("%s", &s[i].f_name); printf("\n Enter Lastname:"); scanf("%s", &s[i].l_name); } for(i=0; i<5; i++) { for(j=i+1; j<5; j++) { if(s[i].roll_no<s[j].roll_no) { temp = s[i].roll_no; s[i].roll_no=s[j].roll_no; s[j].roll_no=temp; } } }					
14	Categorize the basic operations that can be performed on a file with suitable declarations. fopen - open a file- specify how its opened (read/write) and type (binary/text) filepointer=fopen("filename", "mode"); fclose - close an opened file fclose(spData); fread - read from a file size_t fread(void *ptr, size_t size, size_t n, FILE *stream); fwrite - write to a file size_t fwrite(const void *ptr, size_t size, size_t n, FILE*stream); fseek/fsetpos - move a file pointer to somewhere in a file. fseek(FILE *stream, long int offset, int whence) ftell/fgetpos - tell you where the file pointer is located. offset = ftell(file pointer);	**5**	**3**	**5**	**2**	**2.5.2**
Part – c (2x10 = 20 Marks) Instruction : Answer all the questions						
15 a	Demonstrate a C function that will fill an array with a specified value, i.e. every array element should become the specified value. The	**10**	**3**	**3**	**2**	**2.5.2**

	function must have this prototype: int fillArray(int size, int array[], int value); The function should take three parameters: the length of the array, the array itself, and the value to fill the array with. For example, if the specified value was 42, and the array contained the following 6 elements: 3,1, 4, 1, 5, 9.Your function should replace each of those elements with the value 42: 42, 42, 42, 42, 42,42 **ANSWER:** #include <stdio.h> int fillArray(int size, int array[], int value); int main() { int a[10],i,n,fill_value; printf("Enter the number of elements"); scanf("%d",&n); printf("Enter the elements"); for(i=0;i<n;i++) { scanf("%d",&a[i]); } printf("Enter the value to fill"); scanf("%d",&fill_value); printf("Arrays elements before filling\n"); for(i=0;i<n;i++) { printf("%d\t",a[i]); } fillArray(n,a,fill_value); printf("\nArrays elements after filling\n"); for(i=0;i<n;i++) { printf("%d\t",a[i]); } return 0; } int fillArray(int size, int array[], int value) { int i; for(i=0;i<size;i++) { array[i]=value; } }					
	OR					
b	Explain in detail with an example illustrating pointer declaration and dereferencing pointers, Void Pointers and size of Void Pointers. **ANSWER:** **Definition** A pointer is a variable whose value is the address of another variable, i.e., direct address of the memory location. This is done by using unary operator * that returns the value of the variable located at the address specified by its operand. **Declaration:** **Syntax :** Datatype *pointervariable; **Syntax Example** int *ip; /* pointer to an integer */ double *dp; /* pointer to a double */ float *fp; /* pointer to a float */ char *ch /* pointer to a character */ Example: int var = 20; int *ip;	**10**	**3**	**4**	**2**	**2.5.2**

ip = &var;

Reference operator (&) and Dereference operator (*)

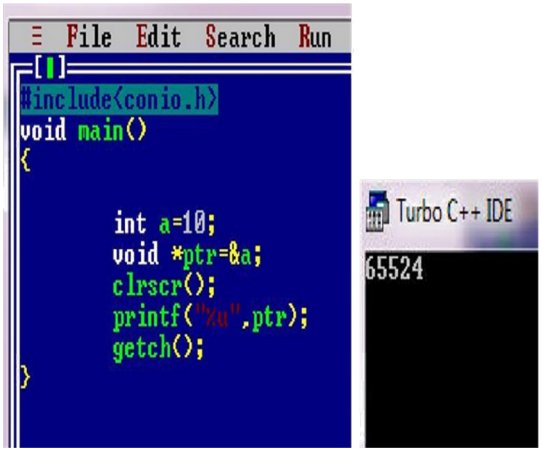
& is called reference operator. It gives the address of a variable.

* is called dereference operator. It gives the value from the address

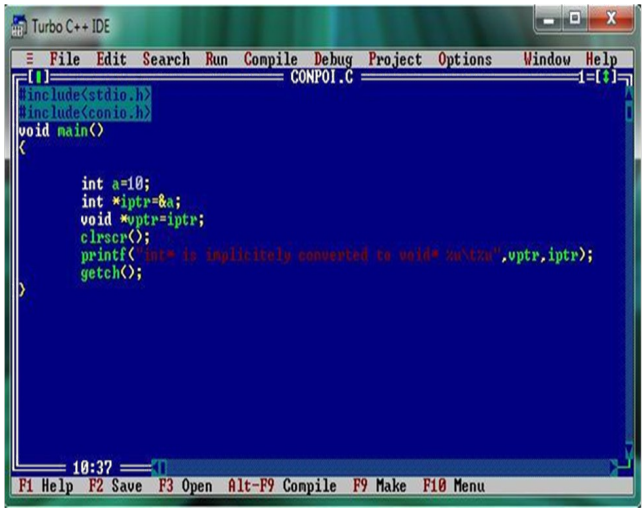
Void Pointer:

1.Void pointer is a generic pointer and can point to any type of object.The type of object can be char, int, float or any other type.

Example



2.A pointer to any type of object can be assigned to a void pointer.



OUTPUT

Size of Void pointer:

The size of void pointer varies system to system. If the system is 16-bit, size of void pointer is 2 bytes. If the system is 32-bit, size of void pointer is 4 bytes. If the system is 64-bit, size of void pointer is 8 bytes.

Here is an example to find the size of void pointer in C language,

	#include <stdio.h> int main() { void *ptr; printf("The size of pointer value : %d", sizeof(ptr)); return 0; }					
16 a	Sejal is working on the menu creation of a library. She wanted to create a structure containing book information like accession number, name of the author, book title, and flag to know whether the book is issued or not. Help her to create a menu in which the following can be done. 1 - Display book information 2 - Add a new book 3 - Display all the books in the library of a particular author 4 - Display the number of books of a particular title 5 - Display the total number of books in the library 6 - Issue a book (If we issue a book, then its number gets decreased by 1 and if we add a book, its number gets increased by 1) #include <stdio.h> #include <string.h> struct book { int an; char title[30]; char author[30]; int issued; }; void display(struct book b) { printf("Accession number-\t%d\nBook-\t%s\nAuthor-\t%s\n",b.an,b.title,b.author); if(b.issued == 0) { printf("Issued-\tNo\n"); } else { printf("Issued-\tYes\n"); } } void add() { //Do yourself //issued will be 0 by defalut } //passing array void	**10**	**4**	**5**	**2**	**2.5.2**

```
book_by_author(struct book *b,int number_of_books,char auth[])
        {
          int i;

for(i=0;i<number_of_books;i++)
          {
            if(strcmp((b+i)->author,auth))
            {

display(*(b+i));
            }
          }
        }

        void book_by_title()
        {
          //do it yourself
        }

        void issue_a_book(struct book b)
        {
        b.issued =1;
        }

        int main()
        {
          //write yourself
          return 0;
        }
```

OR

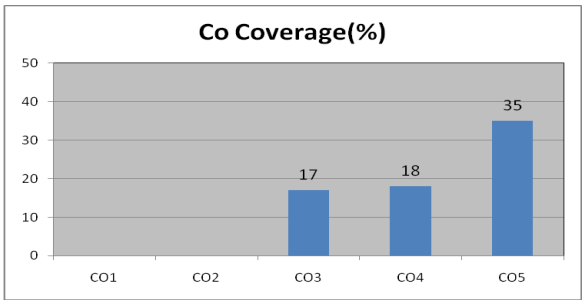
b Assume you need to allocate dynamic memory of 10 byte for storing some information. Say how this can be done. Can the allocated memory be released if not needed? If not released what are the consequences? **10** **4** **5** **2** **2.5.2**

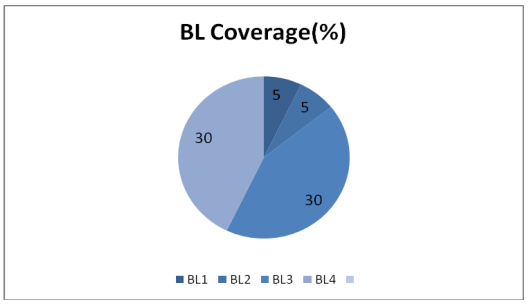
```
int *ptr;
ptr=(type *)malloc(size);
Example program for malloc() function:
#include<stdio.h>
#include<malloc.h>
#include<conio.h>
void main()
{
        float *fp;
        fp=(float *)malloc(10);
        printf("Enter a float value : ");
        scanf("%f", &fp);
        printf("The address of pointer in memory is : %u",
```

	fp); printf("The value stored in memory is : %f", *fp); getch(); } Int *ptr; ptr = (type *)calloc(n,m); Example program 1 for calloc() function: #include<stdio.h> #include<calloc.h> #include<conio.h> void main() { float *fp; fp=(float *)calloc(10,4); printf("Enter a float value : "); scanf("%f", &fp); printf("The address of pointer in memory is : %u", fp); printf("The value stored in memory is : %f", *fp); getch(); } free() function: It is used to release the memory space which is allocated using malloc() or calloc() function Syntax: free(ptr); When program ends, variable ptr goes away, but the space ptr points at does not (allocated on the heap). This is called memory leakage problem.					

***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