aSVKM's NMIMS
Mukesh Patel School of Technology Management & Engineering
Computer Engineering Department
Program: B.Tech Integrated Sem V

Course: Basic Data Structures
2024-2025
Experiment No.03

PART A

(PART A : TO BE REFFERED BY STUDENTS)

A.1 Aim: **To study concepts of Structure, Recursion and Pointers in C.**

Task 1: In a class, there are six students, each having grades for four subjects. Write a program using a 2D array to calculate (i) the total grades for each student and (ii) the average grade for each subject.

Task 2: In a small company there are five salesmen. Each salesman is supposed to sell three products. Write a program using a 2D array to print (i) the total sales by each salesman and (ii) total sales of each item.

Task 3: An industrial plant tracks the energy consumption of five machines across four different shifts. Write a program using a 2D array to find (i) the total energy consumption for each machine and (ii) the total energy consumption for each shift.

Task 4: Create a structure that store details of BankAccount with account number, balance and customer name.

Task 5: Write a C\C++ program-using concept of structures, to store details of BankAccount with account number, balance and customer name. The program must input details from the user and display the details.

Task 6: Modify the program written for task 2. The program must store details for *n* accounts. User gives the value of n. The program must provide a search function that searches details of particular bank account.

Task 7: Write a C\C++ program with recursive functions to perform following

a) Find factorial of a number

Task 8:

Write a program to swap two numbers using concept of pointers.

A.2 Prerequisite:

Knowledge of Recursion and Pointer

A.3 Outcome:

After successful completion of this experiment students will be able to

1. Demonstrate the use of recursion and pointers

A.4 Theory:

Recursive Function:

A recursive function is one which calls itself. This is another complicated idea which you are unlikely to meet frequently. Recursive functions are useful in evaluating certain types of mathematical function. You may also encounter certain dynamic data structures such as linked lists or binary trees. Recursion is a very useful way of creating and accessing these structures.

Advantages and Disadvantages of Recursion

Recursion is more elegant and requires few variables which make program clean. Recursion can be used to replace complex nesting code by dividing the problem into same problem of its sub-type. On other hand,memory usage of recursion is higher as it uses stack memory.

Pointer is variable that contains memory address of another variable. From the definition we can say that the pointer is a variable and you can assign different values of the pointer variable. The value contained by the pointer must be an address that indicates the location of another variable in the memory.

Left Value & Right Value

For instance, after the integer variable x is declared and assigned to a value like this:

```
int x;
X=7;
```

The variable x now has two values:

Left value: 1000

Right value: 7
Where 1000 is the address of the variable x.

The general form of a pointer declaration is
data-type *pointer-name;

The following shows different types of pointers:

char *ptr_c; /* declare a pointer to a character */
int *ptr_int; /* declare a pointer to an integer */
float *ptr_flt; /* declare a pointer to a floating-point */

Example 1:

```
#include <stdio.h>

int main ()
{
  int  var1;
  char var2[10];

  printf("Address of var1 variable: %x\n", &var1  );
  printf("Address of var2 variable: %x\n", &var2  );

  return 0;
}
```

Output -

```
Address of var1 variable: bff5a400
Address of var2 variable: bff5a3f6
```

Example 2:

```
#include <stdio.h>

int main ()
{
  int  var = 20;   /* actual variable declaration */
  int  *ip;        /* pointer variable declaration */

  ip = &var;  /* store address of var in pointer variable*/

  printf("Address of var variable: %x\n", &var  );
```

```
   /* address stored in pointer variable */
   printf("Address stored in ip variable: %x\n", ip );

   /* access the value using the pointer */
   printf("Value of *ip variable: %d\n", *ip );

   return 0;
}
```

Output –

```
Address of var variable: bffd8b3c
Address stored in ip variable: bffd8b3c
Value of *ip variable: 20
```

NULL Pointers in C

It is always a good practice to assign a NULL value to a pointer variable in case you do not have exact address to be assigned. This is done at the time of variable declaration. A pointer that is assigned NULL is called a **null** pointer.
The NULL pointer is a constant with a value of zero defined in several standard libraries. Consider the following program:

```
#include <stdio.h>
int main ()
{
   int  *ptr = NULL;
   printf("The value of ptr is : %x\n", ptr  );
    return 0; }
```

Output –

```
The value of ptr is 0
```

PART B

(PART B: TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Portal)

Roll No.	Name:
Program:	Division:
Semester:	Batch :
Date of Experiment:	Date of Submission:
Grade :	

Task 1:

Task 2:

Task3:

Task 4:

Task 5:

B.5 Conclusion:

(Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.3)

B.6 Question of Curiosity:

Write a C\C++ program with recursive functions to perform following

a) Find the greatest common divisor (GCD) of two integers.
b) Print Fibonacci series up to n term