Laboratory Manual

For

Programming for Problem Solving-I

(ESC201)

B.Tech (IT)

SEM I

July 2021
Faculty of Technology
Dharmsinh Desai University
Nadiad.
www.ddu.ac.in

Table of Contents

LABWORK BEYOND CURRICULA

COMMON PROCEDURE

Tools / Apparatus: Unix/Linux Operating System, Text Editor, gcc compiler

Procedure:

- Prepare the Flow Chart of the Program
- Write the code of the program
- Compile the program for any compile-time errors
- Run the program
- Debug the program for any errors

Sample Experiment

1 AIM: Define a structure for items. The members are item number, item name, item price. Take all the details for at least 5 items. Using function search for the particular item by its name or by its number.

2 TOOLS/APPARATUS: gcc compiler

3 STANDARD PROCEDURES:

3.1 Analyzing the Problem:

- First create a file named as “item.c”.
- After that includes the standard input/output files.
- Now define the structure “item” and its members.
- Create the functions which are necessary for the program.
- Now create the main function and take the information.

3.2 Designing the Solution:

- Create a c file name as “item.c”.
- Define the structure named as “item”. Also define its members that are item number, item name, item price.
- Make a function to search a record by an item number that is “searchitembyno()” or by item name that is “searchitembyname()”.
- Now in the main function declare a variable of the structure item “it[5]”. It should be an array because we want all the details for at least 5 items.
- Now take the all the details by the user using scanf() function.
- Now enter the choice by which you want to search. And call the function for desire output. And display the details using printf() function.

3.3 Implementing the Solution:

3.3.1 Writing Source Code:

```
#include<stdio.h>

struct item                    //defining the structure name item
{
int no;
char name[20];
int price;
};

void searchitembyno(struct item i1[ ] ,int,int);  //function prototype to search by item number

void searchitembyname(struct item i1[ ],char [ ],int); //function prototype to search by item name

void main( )
{
  struct item it[10],t;
  char str[10],c;
  int no,i,j,a,ch;
  printf("\nHow many item :");
  scanf("%d",&n);              // take n number of items
  for(i=0;i<n;i++)
   {
      printf("\nEnter no, name and price of item :");
      scanf("\n%d %s %d",&it[i].i_no,it[i].name,&it[i].price);
 }

label:

// now for searching there are two options
printf("\n\nEnter 1 to search by number and 2 to search by name : ");
scanf("%d",&ch);

switch(ch)
{
case 1:    printf("\n\nEnter the no for the item u want to search : ");
           scanf("\n%d",&no);
           printf("\n");
           searchitembyno(it,a,no); //this is a function call to search by number
           break;
```

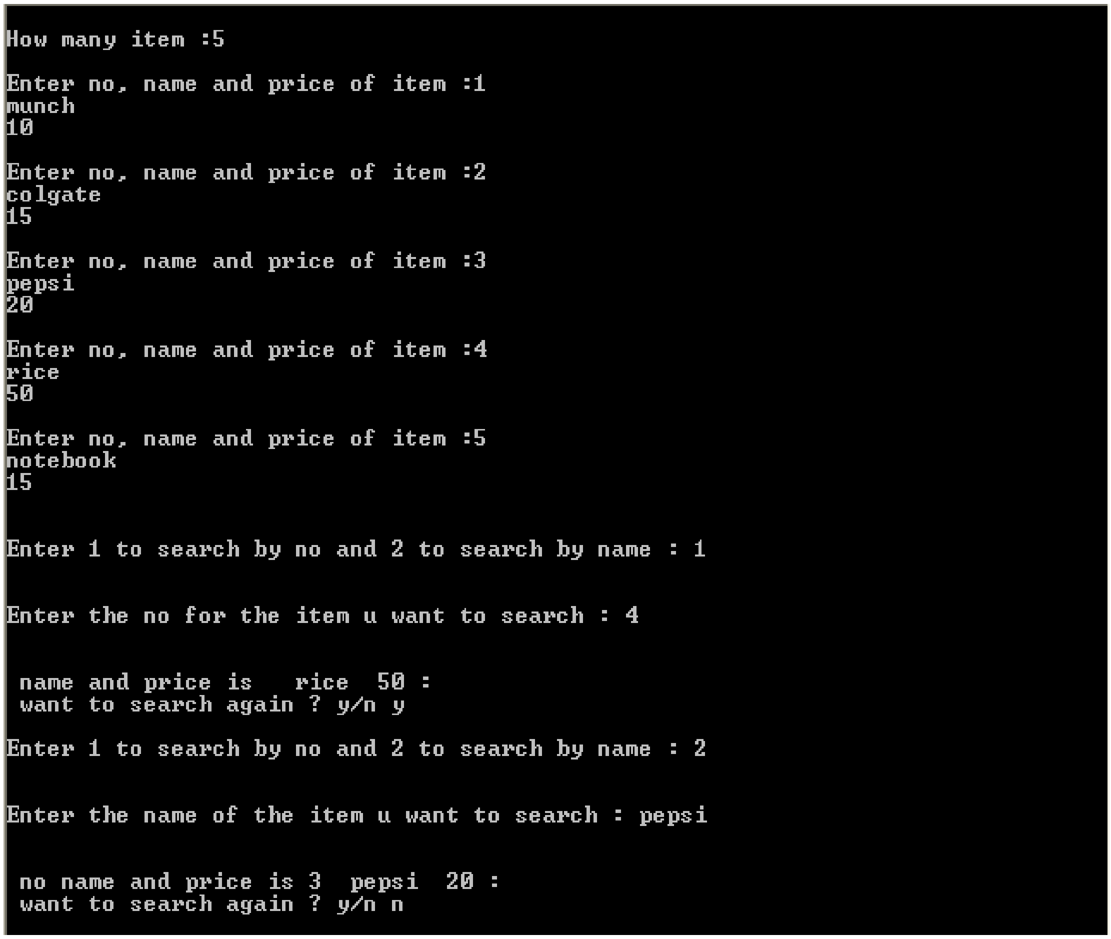
```
case 2:    printf("\n\nEnter the name of the item u want to search : ");
           scanf("\n%s",str);
           printf("\n");
           searchitembyname(it,str,n); //this is a function call to search by name
           break;
default:   break;
}

printf("\n want to search again ? y/n ");  //to search again Enter ‘y’ or to stop searching Enter ‘n’.

c=getchar();
if(c=='y')
{
 goto label ;
}

}
//function definition to search by number
void searchitembyno(struct item i1[],int a,int n)
 {
     int i,flag=0;
     for(i=0;i<n;i++)
     {
         if(i1[i].no==a)      //comparing nos
          {
            flag=1;
            printf("\n name and price is   %s  %d :",i1[i].name,i1[i].price);
            break;
          }
     }

   if(flag==0)
     {
       printf("\nitem not found ");
     }
}

//function definition to search by name
void searchitembyname(struct item i1[],char s[],int n)
  {
     int i,flag=0;
     for(i=0;i<n;i++)
      {
          if((strcmp(i1[i].name,s))==0)    //comparing string
           {
             flag=1;
            printf("\n no name and price is %d  %s  %d :",i1[i].no,i1[i].name,i1[i].price);
            break;
          }
     }

    if(flag==0)
     {
       printf("\nitem not found ");
   }
}
```

3.3.2 Compilation/Running and Debugging the Solution:

- Compile program by using gcc command

  gcc item.c

- If Successful Compilation is done then Run the Code Using
  ./a.out



```
How many item :5

Enter no, name and price of item :1
munch
10

Enter no, name and price of item :2
colgate
15

Enter no, name and price of item :3
pepsi
20

Enter no, name and price of item :4
rice
50

Enter no, name and price of item :5
notebook
15


Enter 1 to search by no and 2 to search by name : 1


Enter the no for the item u want to search : 4


 name and price is   rice  50 :
 want to search again ? y/n y

Enter 1 to search by no and 2 to search by name : 2


Enter the name of the item u want to search : pepsi


 no name and price is 3  pepsi  20 :
 want to search again ? y/n n
```

3.4 Testing the Solution:

- User must have entered all the details with respected to its data type.
- In search by name or number if that record is found than it will display the desire output. Otherwise it will display that item not found.
- If we enter a character instead of integer than the output will be unpredictable.
- Same way if we enter an integer instead of character than the output will be unpredictable.

4 Conclusions:

Hence we have concluded that this experiment will give us the knowledge that how to code a meaningful and understandable program, as well as how to analyze, design and test the program.

Required Software/ Software Tool:

- Linux Operating System
- Terminal (gcc compiler)

Common procedure:

Step 1: For the given problem statement design
Flowchart/Algorithm/Logic
Step 2: Define variables and functions which will show the flow of program
Step 3: Write C code in the file with .c extension
Step 4: Compile code using gcc compiler, which will create a.out executable file.
Step 5: Test the program using sample input and write down output.

EXPERIMENT-1

Aim: Overview of C, gcc compiler and basic program of C

1) Overview of C & GCC compiler.
2) W.A.P to display a “Hello world” message.
3) W.A.P which demonstrate the use \n, \t escape sequences.

Tools: Terminal (gcc Compiler).

Procedure:

1) Give the overview of C, GCC compiler.
2) Make a C program to display the message “Hello World”
3) Make a C program to demonstrate the use \n, \t escape sequences.

EXPERIMENT-2

Aim: Implement the programs using Operators and Expressions.

1) W.A.P to convert the temperature unit from Fahrenheit to Celsius using the formula C = (F – 32) / 1.8.
2) Assume that any month is of 30 days. Now you are given total days. Find out the exact number of Years, Months & Days.
3) You are given time in total seconds. Convert it into Hour: Min: Seconds format.

Tools: Terminal (gcc Compiler).

Procedure:

1) Make a C program in which converts the temperature taken by user in Fahrenheit to Celsius. And for conversion use the formula which is given.
2) Make a program in which the total numbers of days are given and you have to convert that in perfect no of days, year and month.
3) Same as the above program but here instead of days the seconds are given and you need to convert it into hour, min and seconds format.

EXPERIMENT-3

Aim: Implement the programs using Decision making and branching.

1) W.A.P to determine whether input number is ODD or EVEN. Display appropriate message.
2) Write a C program to check if a given Number is Zero or Positive or Negative Using if...else statement.
3) W.A.P that will display Grade of student according to his/her marks using if else ladder

   Marks > 80 then Grade = A
   Marks between 61 & 80 then Grade = B
   Marks between 51 & 60 then Grade = C
   Marks between 41 & 50 then Grade = D
   Marks between 35 & 40 then Grade = E
   Marks < 35 then Grade = F
4) W.A.P which prepare calculator using switch case.

Tools: Terminal (gcc Compiler).

Procedure:

1) Enter the number. And make the coding that if the entered number is odd than display the message the "No is odd". Else display that "No is even".
2) Enter the number and check if a number is Zero or Positive or Negative.
3) Make a C code for grading a student based on the marks. You need to use the if else condition.
4) Prepare basic calculator to perform addition, subtraction, multiplication and division.

EXPERIMENT-4

Aim: Implement the programs using Decision making and looping.

1) W.A.P to print the following patterns.

```
 *                          1
* *                         1 2
* * *                       1 2 3
* * * *                     1 2 3 4
* * * * *                   1 2 3 4 5

    *                               1
   * *                             22
  * * *                           333
 * * * *                         4444
* * * * *                       55555
```

2) W.A.P that computes and prints the Factorial of a given number.
3) W.A.P that computes and prints the Fibonacci series.
4) W.A.P to count Blanks, Tabs and Newlines using while and getchar.
5) W.A.P to check whether the input number is prime or not.

Tools: Terminal (gcc Compiler)

Procedure:

1) Using loops generate all the patterns.
2) Write a program to make the factorial of a given number. In that user enter a number and you have to find the factorial of a given number. (e.g. 5! =5 * 4 * 3 * 2 * 1)
3) Write a program to generate the Fibonacci series for n terms. Where the value for n is given by the user.
4) Write a C code which will count the no of blanks, tabs and newlines in a program, using while and getchar. Here the string is taken by getchar function which is used to get a character from the user.
5) Make a program in which number is entered by user and you need to check that number is prime or not. Divide the number by 2 and at the end if you get 1 than that number is prime.

EXPERIMENT-5

Aim: Implement the programs of Arrays.

1) Find out sum and average of N numbers using array.
2) Multiplication of two 3X3 matrices.
3) Write a program which Sorts all the elements of 1-D Array.
4) Find out the Minimum and Maximum number from the given 1-D array without sorting the array. Also swap position of Minimum and Maximum number.
5) W.A.P which insert element at particular index enter by user.
6) W.A.P which delete particular element enter by user.

Tools: Terminal (gcc Compiler).

Procedure:

1) Create an array for n numbers where the value for n is given by user. Using 1-D array you can take the value and using loop you can find the sum as well as average for n numbers.
2) Same way multiply the two 3x3 matrices based on multiplication rule in matrix.
```
c[i][j]+=a[i][k]*b[k][j]; // using 3 loops
```
3) Apply the logic of sorting for n element in an array. Using two loops compare each and every element.
```
          if(a[j]>a[j+1])
           {
 temp = a[j];
      a[j]=a[j+1];
 a[j+1]=temp;
 }
```
4) Create an array for n numbers where the value for n is given by user. Using 1-D array you can take the value and without sorting find maximum and minimum number.
5) Take the appropriate position or index where you want to insert a new element. After that shift other element down, create the space for new element and insert it.
6) First Search element and then shift all elements up. If element not found then display appropriate message.

EXPERIMENT-6

Aim: Implement the programs of Character Array.

1) W.A.P that will read two strings.
   If given two strings are same then print "Given strings are same".
   If given two strings are not same then print "Given strings are not same"
. (without using strcmp() function)
2) W.A.P which convert all lower case characters of a given string into upper case and all upper case characters of a given string into lower case. Also count number of characters of a given string.
3) W.A.P which searches for the pattern (sub string) in the subject string (main string) and returns the position/index of the start of the string where match is found.
4) W.A.P which check whether given string is palindrome or not.
5) W.A.P which find the reverse of a given string.
6) W.A.P for sorting of strings.

Tools: Terminal (gcc Compiler).

Procedure:

1) Take two strings from the user using loop, after that compare each and every character of one string with another string and display appropriate message.
2) Read one string from user.As per ASCII value compare every character to find lower and upper case and convert it.
3) Take two strings from the user. Consider second string as a pattern and write logic which search a pattern in a subject string and display appropriate message.
4) Take string from the user. Compare first character with last, second character with second last and so on. Print appropriate message.
5) Read one string from user and write logic of reverse of a string.
6) Read table of strings from user and write logic of sorting of a strings.

EXPERIMENT-7

Aim: Write following programs using Functions.

1) Printline() which prints ‘=’ sign 81 times in the same line.
2) Write a function to calculate and display the total amount, take value for principle amount p, rate of interest r and period n from user. Find out the Total_amount using the equation Total_amount = Total_amount * (1+ r) (for number of period n, initialize Total_amount with p)
3) Modify program (above program) for returning total amount
4) Prepare calculator by using concept of function. Create separate functions for addition, subtraction, multiplication and division.

Tools: Terminal (gcc Compiler).

Procedure:

1) Make a function which prints the ‘=’ sign 81 times using a loop.
2) Make functions which calculate the total amount by the formula which is given.
3) Modify the above program which returns the total amount.
4) Create separate functions for addition, subtraction, multiplication and division. Call all four functions form main function and print final answer of all operation.

EXPERIMENT-8

Aim: Implement programs using Recursive function & Arrays as arguments to function.

1) Implement Exponent function using recursive logic. (exp(x,y) = x^y)
2) Implement Sum of digits of a given number using Recursion.
3) Implement sorting of n numbers in ascending order using function (pass array as argument to the function)
4) Read two string from user. Copy last 3 characters of second string to the end of first string by passing strings to the function.

Tools: Terminal (gcc Compiler).

Procedure:

1) Make a recursive function for exponent
   ```
   ans=x*exp(x,y-1);  //exp is a function
   ```
2) Make a a recursive function for sum of digits of a given number.
   ```
   ans=d+dsum(n);
   ```
3) Make a program which sort the given numbers using function in which array is given as an argument. Repeatedly swap the adjacent elements if they are in wrong order.
4) Read two string and pass both the strings as an arguments to the function. Find out length of second string and copy last 3 characters to the end of first string.

EXPERIMENT-9

Aim: Implement the programs using Structures.

1) Generate a result table which consists of student id, student name, marks of three subject and total marks. Write a program which takes input for ten students and displays result table. Also display student information separately who got the highest total.
2) Create a structure containing information of products viz, id, name, price etc. Implement search function on basis of id of product and displays its respective name and price.
3) Modify above program and sort product details according to name(alphabetically order)
4) Create a structure item with three members name, price and quantity. Read information of one item. After reading information of one item update price and quantity of that item by calling update function with necessary arguments. Finally print updated value of item.

Tools: Terminal (gcc Compiler).

Procedure:

1) Design a structure with members as student id, student name, marks of three subject and total marks.
   Using loop take information for ten students and displays result table.
   Apply logic to get highest total marks and display all information.
2) Create a structure for product with members as id, name, and price.
   Take id as input and search a product from the list of product and display all information.
3) Modify the above program, apply the logic of sorting and swap the whole structure and display all the details.
4) Implement the program by passing a copy of the structure to the function.

EXPERIMENT-10

Aim: Implement the programs using Union.

1) Read and display the members of Union (Take book details-ISBN no , Title, Price)
2) Store information of 10 persons. Information includes name and age. But criteria is for the child (age<=18) display age in form of full birth date(dd-mm-yyyy), for an adult the age should be in years only(Current year-Birth year), while for aged person(age>60) display message “old person”.

   Note:- Consider age= Current year-Birth year

Tools: Terminal (gcc Compiler).

Procedure:

1) Design Union with members as book ISBN no, title and price. Take the information from user and display it.
2) Design Union for person details. Take the input for 10 people and display the information as described.

EXPERIMENT-11

Aim: Implement the programs using Pointers.

1) Swap/exchange values of two integer variables using function. Use pointers.
2) Write a function mystrcat(s,t) which copies string t to the end of the strings.
3) Addition of two 3X3 matrices using pointers.
4) W.A.P that will find maximum number from the given 1-D array by using the concept of function returning pointer.
5) WAP to check the given nxn matrix is symmetric or not using pointer.
6) WAP to store data for n number of student (i.e. student name, id and percentage) and display the detail of a student who got the highest percentage. Implement the program using the concept of pointers and structures.

Tools: Terminal (gcc Compiler).

Procedure:

1) Exchange value of two variable using pass by address method.
   Create a separate function swap() with two integer pointers as arguments.

```
 void swap (int *a1 , int * b1)
{    temp=*a1;
    *a1=*b1;
    *b1=temp;
 }
```

2) Create mystrcat(char *s ,char *t) function, append the string t at the end of string s.
3) Create two 3X3 array, using pointer add the corresponding elements in array.
   *(p3+i*3+j) = *(p1+i*3+j) + *(p2+i*3+j) ; // p1=&a, p2=&b, p3=&c, where a, b and c are 3x3 array and p1,p2 and p3 are the integer pointers.
4) Read 1-D array from user.Write a logic to find out maximum number. Return address of maximum number to the calling function.
5) A symmetric matrix is a matrix whose transpose is equal to the matrix itself. Compare matrix and transpose of a matrix and display appropriate message.
6) Take any structure, assigned a pointer for that structure and using the pointer take details and display detail of a student who got the highest percentage it. (access the structure member using -> (Arrow operator) through pointer variable)

LABWORK BEYOND CURRICULA

EXPERIMENT 12:

- Overview of Dynamic Memory allocation and Program using Dynamic Memory allocation functions.

EXPERIMENT 13:

- Overview of Pre-Processor directives and Program using Pre-Processor directives.