Jawaharlal Nehru Engineering College

Laboratory Manual

PIC

For

F.Y. (M.C.A) Students

16, Oct 2009 – Rev 00 – MCA – ISO 9000 Tech Document

FORWARD

It is my great pleasure to present this laboratory manual for FY (MCA) students for the subject of PIC (Programming In C). Subject name itself is expecting programming skills that are needed as basis of almost all programming language.

As a student, many of you may be wondering with some of the questions in your mind regarding the subject and exactly what has been tried is to answer through this manual.

As you may be aware that MGM has already been awarded with ISO 9000 certification and it is our endure to technically equip our students taking the advantage of the procedural aspects of ISO 9000 Certification.

Faculty members are also advised that covering these aspects in initial stage itself, will greatly relived them in future as much of the load will be taken care by the enthusiasm energies of the students once they are conceptually clear.

Dr. H. H. Shinde
Principal

LABORATORY MANNUAL CONTENTS

This manual is intended for the students FY MCA for the subject of PIC (Programming In C). Subject name itself is expecting what are basic and the advance skills that might be requires as a basis of a programming language. In the subject of this manual typically contains practical/Lab Sessions we have dealt with C programming language. i.e. C language (data types, operators, arrays, structures and files.

Students are advised to thoroughly go through this manual rather than only topics mentioned are the key to understanding and conceptual visualization of theoretical aspects covered in the books.

Good Luck for your Enjoyable Laboratory Sessions

Prof. N.R. Yadav

SUBJECT INDEX:

1. Study of if-else control structure
2. Study of switch-case control structure
3. Study of while control structure
4. Study of for control structure
5. Study of functions
6. Study of recursive functions
7. Study of arrays.
8. Study of sorting techniques.
9. Study of 2-dimensional arrays.
10. Study of structures
11. Study of files.

DOs and DON'T DOs in Laborary:

1. Do not handle any equipment before reading the instructions/Instruction manuals

2. Read carefully the power ratings of the equipment before it is switched on whether ratings 230 V/50 Hz or 115V/60 Hz. For Indian equipments, the power ratings are normally 230V/50Hz. If you have equipment with 115/60 Hz ratings, do not insert power plug, as our normal supply is 230V/50 Hz, which will damage the equipment.

3. Observe type of sockets of equipment power to avoid mechanical damage

4. Do not forcefully place connectors to avoid the damage

5. Strictly observe the instructions given by the teacher/Lab Instructor

Instruction for Laboratory Teachers:

1. Submission related to whatever lab work has been completed should be done during the next lab session. The immediate arrangements for printouts related to submission on the day of practical assignments.

2. Students should be taught for taking the printouts under the observation of lab teacher.

3. The promptness of submission should be encouraged by way of marking and evaluation patterns that will benefit the sincere students.

WARMUP EXCERCISES:

What is software and hardware?

What is a flowchart?

What is an algorithm?

What is a program?

What is a programming language?

How programming language is different than a software package?

What are different data types in C?

What are the rules to define identifier and a constant?

Lab Exercises:

Exercise No1: (2 Hours) – 1 Practical

If-else control structure:

Write a program using if-else control structure to find roots of quadratic equation, $ax^2+bx+c=0$

Input: Values of a, b and c.

Output: Roots of quadratic equation.

Algorithm:

```
Step 1: Read values of a, b and c.
Step 2: Calculate d= b*b -4*a*c
Step 3: If d > 0 then
        Step 3.1: Print that the roots are real and unequal.
        Step 3.2: Calculate r1 = (-b + sqrt (d)) / (2*a).
        Step 3.3: Calculate r1 = (-b - sqrt (d)) / (2*a).
        Step 3.4: Display r1 and r2.
        Step 3.5: stop.
Step 4: If d == 0 then
        Step 4.1: Print that the roots are real and equal.
        Step 4.2: Calculate r1 = (-b + sqrt (d)) / (2*a).
        Step 4.3: Calculate r2 =r1.
        Step 4.4: Display r1 and r2.
        Step 4.5: stop.
Step 5: If d > 0 then
        Step 5.1: Print that the roots are imaginary.
        Step 5.2: Calculate d= -d.
        Step 5.2: Calculate real_part = (-b)/ (2*a).
        Step 5.3: Calculate img_part = (sqrt (d)) / (2*a).
        Step 5.4: Display r1 as real_part + img_part "i"
        Step 5.6: Display r2 as real_part - img_part "i"
        Step 5.7: Stop.
```

Exercise No 2: (2 Hours) – 1 Practical

Switch control structure:

Write a program using Switch control structure to find roots of quadratic equation, $ax^{2}+bx+c=0$

Input: Values of a, b and c.

Output: Roots of quadratic equation.

Algorithm:

```
  Step 1: Read values of a, b and c.
  Step 2: Calculate d= b*b -4*a*c
  Step 3: If d > 0 then ch=1, if d=0 then ch = 2, if d<0 then ch = 3
  Step 4: If ch == 1 then
        Step 4.1: Print that the roots are real and unequal.
        Step 4.2: Calculate r1 = (-b + sqrt (d)) / (2*a).
        Step 4.3: Calculate r1 = (-b - sqrt (d)) / (2*a).
        Step 4.4: Display r1 and r2.
        Step 4.5: stop.
Step 5: If ch == 2 then
        Step 5.1: Print that the roots are real and equal.
        Step 5.2: Calculate r1 = (-b + sqrt (d)) / (2*a).
        Step 5.3: Calculate r2 =r1.
        Step 5.4: Display r1 and r2.
        Step 5.5: stop.
Step 6: If ch == 3 then
        Step 6.1: Print that the roots are imaginary.
        Step 6.2: Calculate d= -d.
        Step 6.2: Calculate real_part = (-b)/ (2*a).
        Step 6.3: Calculate img_part = (sqrt (d)) / (2*a).
        Step 6.4: Display r1 as real_part + img_part "i"
        Step 6.6: Display r2 as real_part - img_part "i"
        Step 6.7: Stop.
```

Exercise No 3: (2 Hours) – 1 Practical

While control structure:

Write a program using while control structure to find sum of factorials.

Input: n (number of elements)

Output: Sum of 1! + 2! + 3! + …… +n!

Algorithm:

```
Step 1: Read value of n.
Step 2: Initialize i=1, j-1, fact=1 and sum =0.
Step 3: if (i<=n) then do steps 4 else go to step 8
Step 4: if (j <= i) then
        Step 4.1: Calculate fact = fact * j.
        Step 4.2: increment j.
        Step 4.3: go to step 4
        Else go to step 5
 Step 5: calculate sum = sum + fact.
 Step 6: Increment i.
 Step 7: Go to step 4.
 Step 8: Display sum.
 Step 9. Stop.
```

Exercise No 4: (2 Hours) – 1 Practical

For control structure:

Write a program using for control structure to print Fibbonacii series.

Input: n (number of terms)

Output: 1,1,2,3,5,8,13,21 ……. Such n terms.

Algorithm:

```
Step 1: Read value of n.
Step 2: Initialize f1=1, f2=1
Step 3: Display f1 and f2.
Step 4: Repeat step 4 to 8, n number of times using for loop.
Step 5: Calculate f3= f1+f2.
Step 6: Display f3.
Step 6: Assign f1=f2.
Step 7: Assign f2=f3.
Step 8. Stop.
```

Exercise No 5: (2 Hours) – 1 Practical

Functions

Write a program using Function to generate Fibbonacii Series

Input: n (number of terms)

Output: 1,1,2,3,5,8,13,21 ……. Such n terms.

Algorithm:

```
Step 1: Read value of n.
Step 2: Initialize f1=1, f2=1
Step 3: Display f1 and f2.
Step 4: Call function fibbo with n, f1 and f2 as parameters passed to it.
Step 5. Stop.
Function fibbo (ln, lf1, lf2)
Step 1 : Repeat step 2 to 6, ln number of times using for loop.
Step 2: Calculate f3= lf1+lf2.
Step 3: Display f3.
 Step 4: Assign lf1=lf2.
 Step 5: Assign lf2=f3.
 Step 6. Return.
```

* note l prefix is used to denote local variable.

Exercise No 6: (2 Hours) – 1 Practical

Recursive Functions

Write a program using recursive Function to calculate factorial of number.

Input: n (number of terms)

Output: Factorial of n

Algorithm:

```
Step 1: Read value of n.
Step 2: Call function fact with n as parameters passed to it.
Step 3: Display factorial.
Step 4: Stop.

Function fact (int m)
Step 1: if (m==1) then return 1
Step 2: if (m! =1) then call fact function with (n-1) as parameter passed to it.
Step 3: Return (fact)
```

* note l prefix is used to denote local variable.

Exercise No 6: (2 Hours) – 1 Practical

Recursive Functions

Write a program to find minimum number from an array.

Input: n (size of array), array elements.

Output: Minimum number.

Algorithm:

```
Step 1: Read value of n.
Step 2: Read n number of array elements (using for loop).
Step 3: Display array.
Step 4: Assume first element of array as the minimum element
        i.e. min = a[0] and loc= 0.
Step 5: Repeat step 6, n number of times.
Step 6: If a[i] < min then
        Step 6.1: Assign min = a[i].
        Step 6.2: Assign loc = i.
Step 7: Display minimum number and its location.
Step 8: Stop.
```

Exercise No 7 : (2 Hours) – 1 Practical

One dimensional Array, Selection sort

Write a program to sort an array using selection sort technique.

Input: n (size of array), array elements.

Output: Sorted array.

Algorithm:

```
Step 1: Read value of n.
Step 2: Read n number of array elements (using for loop).
Step 3: Display array.
Step 4: Repeat step 5 to 9 ,ie for j=1 to (n-1) number of times
Step 5 : Let loc= j;
Step 6: Let min = a[j]
Step 7 :Repeat step 8 ie for I = (j+1) to n number of times
Step 8: If a[j] < min then
        Step 8.1 : min = a[j]
        Step 8.2 : loc= j.
Step 9: Swap a[loc] and a[i]
Step 10: Display sorted array.
Step 11. Stop.
```

Exercise No 8 : (2 Hours) – 1 Practical

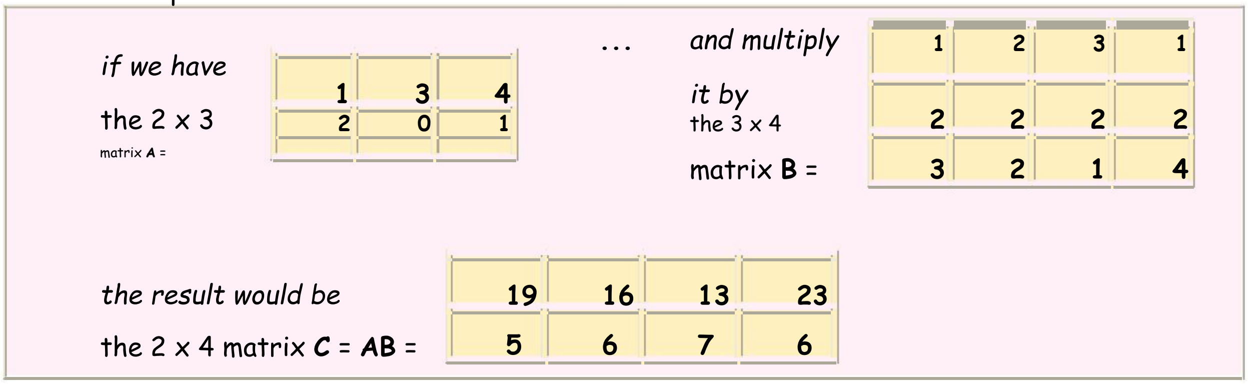
Two dimensional Array, Matrix multiplication program

Write a program to multiply two metrices.

Input: Martix a, Matrix b.

Output: Multiplication of a and b = matrix c.

Theory : GIVEN: Two arrays, **A**, which is *m* x *n*, and **B**, which is *n* x *p*.
The Product **AB** is determined as the dot products of the i^{th} row in **A** and the j^{th} column in **B**, placed in i^{th} row and j^{th} column of the resulting *m* x *p* matrix **C**.
For example:



if we have the 2 x 3 matrix **A** = $\begin{matrix} 1 & 3 & 4 \\ 2 & 0 & 1 \end{matrix}$ ... *and multiply it by* the 3 x 4 matrix **B** = $\begin{matrix} 1 & 2 & 3 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 2 & 1 & 4 \end{matrix}$

the result would be the 2 x 4 matrix **C** = **AB** = $\begin{matrix} 19 & 16 & 13 & 23 \\ 5 & 6 & 7 & 6 \end{matrix}$

Algorithm:

```
Step 1: Read value of m and n.
Step 2: Read value of n and p
Step 3 if (m != n ) display error and stop.
Step 4: Read matrix a (using nested for loop).
Step 5: Read matrix b (using nested for loop).
Step 6: For i = 0 to n-1 repeat step 7 to 11
Step 7: for j = 0 to p-1 repeat step 8 to 11
Step 8: Initialise sum = 0
Step 9: For k = 1 to n do step 10
Step 10: Calculate sum = sum + a[I,k]* b[k.j];
Step 11: Set c[i,j] = sum.
Step 12: Display matrix c.
Step 13: Stop
```

Exercise No 9 : (2 Hours) – 1 Practical

Structure in C

Write a program to accept and display information of students using structures.

Input: Student information.

Output: Student information

Theory: Structure is a composition of variables if different types, grouped together under a single name. Each variable within a structure is called a member, name given to structure is called as a tag. Data type of variables can be any of C data types including arrays, pointers and a structure.

Syntax to declare a structure is:

```
structure tag {
        data_type member1;
        data_type member2;
        ……
        data_type membern; }
```

Algorithm :

Step 1: Declare a structure.
Step 2: Accept student information.
Step 3: Display student information.
Step 4. Stop

Exercise No 10 : (2 Hours) – 1 Practical

File handling in C

Write a program to accept and display information of students using files

Input: Student information.

Output: Data file containing Student information.

Theory: C provides various functions to handle files. These finctions are:

a) File Open
b) File Read
c) File write
d) File Close
e) Enf of file marker,   etc

To access any file it must be opened using fopen(). We can read from or write in a file using fread() and fwrite() functions respectively. After using a file we must close it using fclose() function.

Algorithm :

```
Step 1: Declare a file.
Step 2: Open a file in write mode using fopen().
Step 3: Accept student information.
Step 4. Write information in a file using fwrite().
Step 5. Close file.
Step 6: Open the file in read mode using fopen()
Step 7: Read file contents using fread()
Step 8: Repeat step 7 till end of file.
Step 9: Close the file.
Step 10: Stop.
```

Quiz on the subject:

Quiz should be conducted on tips in the laboratory, recent trends and subject knowledge of the subject. The quiz questions should be formulated such that questions are normally are from the scope outside of the books. However twisted questions and self formulated questions by the faculty can be asked but correctness of it is necessarily to be thoroughly checked before the conduction of the quiz.

Conduction of Viva-Voce Examinations:

Teacher should oral exams of the students with full preparation. Normally, the objective questions with guess are to be avoided. To make it meaningful, the questions should be such that depth of the students in the subject is tested Oral examinations are to be conducted in co-cordial environment amongst the teachers taking the examination. Teachers taking such examinations should not have ill thoughts about each other and courtesies should be offered to each other in case of difference of opinion, which should be critically suppressed in front of the students.

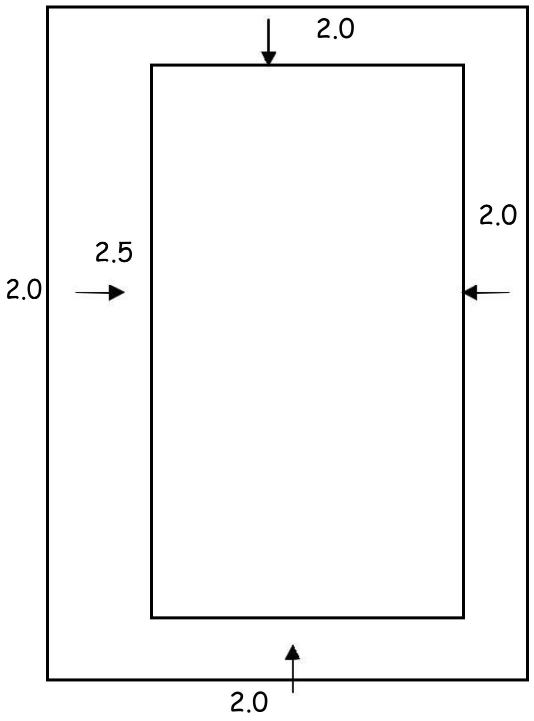
Submission:

Document Standard:

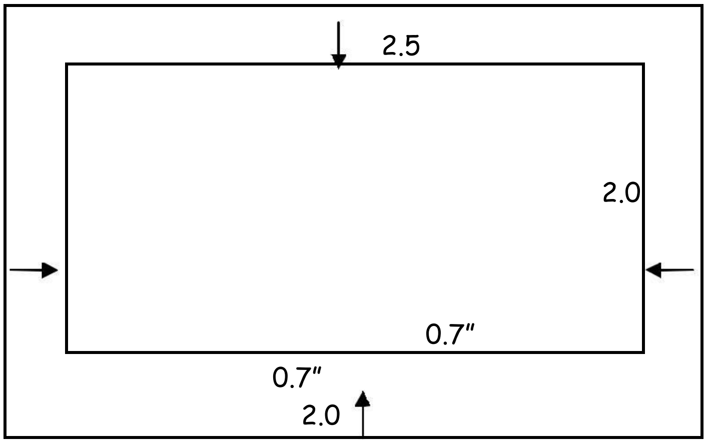
A] Page Size	A4 Size
B] Running text	Justified text
C] Spacing	1 Line

D] Page Layout and Margins (Dimensions in Cms)

Normal Page



Horizantal



Desription	Font	Size	Boldness	Italics	Underline	Capitalize
College Name	Arial	24	-----	------	Yes	----------
Document Title	Tahoma	22	-----	------	---------	----------
Document Subject	Century Gothic	14	-----	------	---------	Capital
Class	Bookman old Slyle	12	-----	------	---------	----------
Document No	Bookman old Slyle	10	-----	------	---------	----------
Copy write inf	Bookman old Slyle	9	-----	------	---------	----------
Forward heading	Bookman old Slyle	12	-----	------	Yes	Capital
Forward matter	Bookman old Slyle	12	-----	------	---------	----------
Lab man Contents title	Bookman old Slyle	12	-----	------	Yes	Capital
Index title	Bookman old Slyle	12	Yes	------	Yes	Capital
Index contents	Bookman old Slyle	12	-----	------	---------	----------
Heading	Tahoma	14	Yes	Yes	Yes	----------
Running Matter	Comic Sans MS	10	-----	------	---------	----------

Evaluation and marking system:

Basic honesty in the evaluation and marking system is absolutely essential and in the process impartial nature of the evaluator is required in the examination system to become popular amongst the students. It is a wrong approach or concept to award the students by way of easy marking to get cheap popularity among the students to which they do not deserve. It is a primary responsibility of the teacher that right students who are really putting up lot of hard work with right kind of intelligence are correctly awarded.

The marking patterns should be justifiable to the students without any ambiguity and teacher should see that students are faced with unjust circumstances.