Semester I/II

Computer Programming Lab Manual

Session : Aug 2014 - Nov 2014

Sub Code: 14CPL 16 / 14CPL26

PART – A : Demonstration of Personal Computer and its Accessories Demonstration and Explanation on Disassembly and Assembly of a Personal Computer by the faculty-in-charge. Students have to prepare a write-up on the same and include it in the Lab record and evaluated.

Laboratory Session-1: Write-up on Functional block diagram of Computer, CPU, Buses, Mother Board, Chip sets, Operating System & types of OS, Basics of Networking & Topology and NIC.

Laboratory Session-2: Write-up on RAM, SDRAM, FLASH memory, Hard disks, Optical media, CD-ROM/R/RW, DVDs, Flash drives, Keyboard, Mouse, Printers and Plotters.

Note: These **TWO Laboratory sessions** are used to fill the gap between theory classes and practical sessions.

PART – B: Problem Solving in C

http://www.c4learn.com/c-programs/category/file-programs

Problem Statement 1 :

Design and develop a flowchart or an algorithm that takes three coefficients (***a***, ***b***, and ***c***) of a Quadratic equation ($ax^2+bx+c=0$) as input and compute all possible roots. Implement a C program for the developed flowchart/algorithm and execute the same to output the possible roots for a given set of coefficients with appropriate messages.

Objective:

To understand the decision making constructs and find the roots of a quadratic equation

Algorithm:

```
Step 1.[input the values of  a,b,c]
        read a,b,c

Step 2.[check if a or b are zero]
        if(a=0 or b=0)
           output "invalid input"
        goto step 7

Step 3.[calculate the discriminant]
        d=b*b - 4ac

Step 4.[find two distinct roots]
         if(d>0)
                r1=(-b+sqrt(d))/2a
                r2=(-b-sqrt(d))/2a
        Output r1 and r2 with suitable message
         goto step 7

Step 5. [find two equal roots]
```

```
        if(d=0)
                r1=r2=-b/2a
                Output r1 and r2 with suitable message
        goto step 7

Step 6. [find two complex roots]
                p1= -b/2a
                p2=sqrt(-d)/2a
        Output r1 and r2 with suitable message

Step  7: Stop
```

Problem Statement 2 :

Design and develop an algorithm to find the ***reverse*** of an integer number **NUM** and check whether it is PALINDROME or NOT. Implement a C program for the developed algorithm that takes an integer number as input and output the reverse of the same with suitable messages. Ex: Num: **2014**, Reverse: **4102**, Not a Palindrome

Objective:

To understand the while loop construct and find if a given number is palindrome

Algorithm:

```
Step 1.[input the number]
        Read number

Step 2.[make a copy of the number]
        temp=number

Step 3.[initialize the value of reverse]
        reverse=0

Step 4.[reverse the given number]
        while (temp! =0)
                remain=temp%10
                temp=temp/10
                reverse=reverse*10+remain;
        end while

Step 5.[check if the given number is palindrome]
        if(number = reverse)
                output "given number is palindrome"
        else
          output "given number is not palindrome"
        endif

Step 6.[finished]
        Stop
```

Problem Statement 3:

3a. Design and develop a flowchart to find the square root of a given number ***N***. Implement a C program for the same and execute for all possible inputs with appropriate messages. Note: **Don't use library function *sqrt(n)*.**

Objective:

To understand the while loop construct and find square root of a given number.

Algorithm:

```
Step 1.[input the number]
        Read number
Step 2.Initialise variable LG=1

Step 3.[Call function sqareroot ]
        NG=Sqareroot(LG,NG,n)

Step 4.display result NG

Step 5.STOP

Squareroot(LG,NG,n)

Step 1.[Loop]
        do{
                NG=(0.5*(LG+n/LG));
                LG=NG;
        }while((NG*NG-n)<0.005;

Step 2. return NG.
```

3b. Design and develop a C program to read a ***year*** as an input and find whether it is ***leap year*** or not. Also consider end of the centuries.

Objective:

To understand the use if ladder and find if a year is leap or not.

Algorithm:

```
Step 1.[input the Year]
        Read number

Step 2.[Check if year is leap or Not]
        If(year%400=0]
                Output :year is leap
        Else if(year%100=0)
                Output :year is leap
        Else if(year%4=0)
                Output :year is leap
        Else
                Output:year is not leap
Step 3.Stop
```

Problem Statement 4 :

Design and develop an algorithm for evaluating the polynomial $\mathbf{f(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0}$, for a given value of ***x*** and its coefficients using Horner's method.
Implement a C program for the developed algorithm and execute for different sets of values of coefficients and ***x.***

Objective:

To evaluate a polynomial using Horner's method

Algorithm:

```
Step 1.[input n]
        Read n

Step 2.[Input n coefficients,starting from nth coefficient to 0th coefficient]
        for i =n down to 0 in step -1
                read a[i]
        end for

Step 3.[input the value of x]
        read x

Step 4. [initialize]
         product=a[n]*x

Step 5.[evaluate the polynomial using horner's method]
         for i=n-1 down to 1 in step -1
                product=(product+a[i])*x
        end for
        result=product+a[0]

Step 6.[output the value of the given polynomial expression]
        output result

Step 7.[finished]
        Stop
```

Problem Statement 5 :

Write C Program to compute **Sin(x)** using Taylor series approximation given by
Sin(x) = x - (x^3/3!) + (x^5/5!) - (x^7/7!) + …….
Compare the result with the built- in Library function and print both the results.

Objective:

To understand the implementation of sin(x) series using Taylor series

Algorithm:

```
Step 1: Input x value
        Term=pow(x,1)/factorial(1)

Step 2:[Loop to find term and sum it up]
        For(i=3 ; i<=8;i+2)
        If(term>0)
        Term=-(pow(x,i)/factorial(i))
        If(term<0)
        Term=pow(x,i)/factorial(i)
        Sum+=term

Step 3:display sum and value of standard function sin(x)

Step 4:Stop

Factorial(n)
Step 1:if(n==0 || n==1)
                Return 1
        Else
                Return(n*factorial(n-1)
Step 2 :Stop
```

Problem Statement 6:

Develop, implement and execute a C program that reads ***N*** integer numbers and arrange them in ascending order using ***Bubble Sort*** technique. Extend the program to perform a search operation on these sorted numbers by accepting a key element from the user applying ***Binary Search*** method. Report the result SUCCESS or FAILURE as the case may be.

Objective:

To understand technique using bubble sort and working of binary search and search for a given key integer in an array.

Algorithm:

```
Step 1.[input the size of the array]
        read n

Step 2.[input the elements of the array ]
        for i= 0 to n-1 in step 1
             read a[i]
        end for

Step 3.[sort the elements using bubble sort
technique]
      for i=0 to n-1 in array a
        for j=n-1 to i in array a
             if(a[j]<a[j-1])
               t=a[j]
               a[j]=a[j-1]
               a[j-1]=t
             end if
        end for

Step 4.[output the sorted elements]
      for i=0 to n-1 in step 1
             output a[i]
      end for

Step 5.[input the key element which has to be
searched]
        read key

Step 6.[initiailze value of low and high]
        low=0
```

```
            high=n-1

Step 7.[conduct binary search to find the given
key element in the array]
            while(low<=high)
              mid=(low+high)/2
              if(key=a[mid])
                    output "successful search,the key
                    is found in position mid+1"
                    stop
              else if(key<a[mid])
                    high=mid-1
             else
        low=mid+1
end if
end while
 output"unsucessful search,the key is not found"
 stop
```

Problem Statement 7:

Develop, implement and execute a C program that reads two matrices ***A*** (**m x n**) and ***B*** (**p x q**) and Compute the product ***A*** and ***B***. Read matrix ***A*** in row major order and matrix ***B*** in column major order. Print both the input matrices and resultant matrix with suitable headings and in matrix format. Program must check the compatibility of orders of the matrices for multiplication. Report appropriate message in case of incompatibility.

Objective:

To find the product of two matrices

Algorithm:

```
Step 1.[input the no. of rows and columns of matrix a]
    read m,n

Step 2.[input the no. of rows and columns of matrix b]
    read p,q

Step 3.[check the compatibility for multiplication of matrices a and b]
    if(n!=p)
        output matrix multiplication is not possible
        goto step 8
    else
        goto step 4
    end if

Step 4.[input the elements of matrix a]
    for i=0 to m-1 in step 1
        for j=0 to n-1 in step 1
            read a[i][j]
        end for
    end for

Step 5.[input the elements of matrix b]
    for i=0 to p-1 in step 1
        for j=0 to q-1 in step 1
```

```
                read b[i][j]
            end for
        end for

Step 6.[perform multiplication of matrices a and b]
        for i=0 to m-1 in step 1
            for j=0 to q-1 in step 1
                sum=0
                for k=0 to n-1 in step 1
                    sum=sum+a[i][k]*b[k][j]
                end for
                c[i][j]=sum
            end for
        end for

Step 7.[output the matrix a]
        for i=0 to m-1 in step 1
            for j=0 ton-1 in step 1
                output a[i][j]
            end for
        end for
Step 8.[output the matrix b]
        for i=0 to p-1 in step 1
            for j=0 to q-1 in step 1
                output b[i][j]
            end for
        end for

Step 9.[output the resultant matrix c]
        for i=0 to m-1 in step 1
            for j=0 to q-1 in step 1
                output c[i][j]
            end for
        end for

Step 8.[Finished]
        Stop
```

Problem Statement 8 :

Write and execute a C program that
i. Implements string copy operation ***STRCOPY***(str1,str2) that copies a string *str1* to another string *str2* without using library function.
.

Objective:

To understand the implementation of library function strcpy()

Algorithm:

```
Step 1. [Input a String]
        gets(s1)

Step 2. [copy character by character in string s1 to string s2]
        while(s1[i]!='\0')
                s2[i]=s1[i];
                i++;
        end while
        s2[i]='\0';

Step 3. Display copied string s2

Step 4. Stop
```

ii. Reads a *sentence* and prints frequency of each of the vowels and total count of consonants

Objective:

To understand the use of counter and || operator

Algorithm:

```
Step 1. [Input a Sentence]
        gets(s1)

Step 2.[Count number of vowels and consonents]
        For(i=0 to s1[i]!='\0')
            if ((sentence[i] == 'a' || sentence[i] == 'e' || sentence[i] ==
                        'i' || sentence[i] == 'o' || sentence[i] == 'u') ||
                        (sentence[i] == 'A' || sentence[i] == 'E' || sentence[i] ==
                        'I' || sentence[i] == 'O' || sentence[i] == 'U'))
                         vowels = vowels + 1;

                else if (sentence[i] =='t' ||sentence[i] =='\0' || sentence[i] ==' ')
                        special = special + 1;
                else
                        consonants = consonants + 1;
                end of if
        end of for

 Step 3.Display number of vowels and consonents

 Step 4.Stop
```

Problem Statement 9 :

a. Design and develop a C function ***RightShift****(x ,n)* that takes two integers *x* and *n* as input and returns value of the integer *x* rotated to the right by *n* positions. Assume the integers are unsigned. Write a C program that invokes this function with different values for *x* and *n* and tabulate the results with suitable headings.

Objective:

To understand the working of right shift operator and concept of functions

Algorithm

```
Step1: [Read the number and the number of bits to be shifted] read x,
        n

Step2: [Call the function rightrot() passing x and n as the parameters]
       res=rightrot(x,n)

Step3: [Print the result]
        Output res

Step4:  [finished]
        stop

rightrot(int x, int n)
Step 1: [right shift x by n bit positions]
            for i=0 to n1-1 in step1
                if x%2 then
                    x1=x1>>1
                    x1+=1<<15
                else
                    x1=x1>>1
                end if

Step 2: [return the value of x]
       return x
```

b. Design and develop a C function ***isprime***(num) that accepts an integer argument and returns 1 if the argument is prime, a 0 otherwise. Write a C program that invokes this function to generate prime numbers between the given range.

Objective:

To understand the concept of functions and check if a given number is prime or not

Algorithm:

```
Step1: [Read the number]
        read n

Step 2: [Call isprime() passing n as the parameter]
               ans = isprime(n)

Step 3:  if ans == 0
               Print “the number is not prime”
         else
               Print “the number is prime”

Step 4: [Read the choice]
        read ch

Step 5: if ch==1
               goto step 1
        else
               goto step 6
        end if

Step 6: Stop

isprime(n) function

Step 1:determine number is prime or not
        for i= 2 to n/2 in step1
          if((n/i)*i ==n)
               return 0;
          end if
        end for
return 1
```

Problem Statement 10:

Develop a function in C called ***MatchAny***(s1,s2) that takes two string arguments and does the following task:
i) if s1 is **substring** of s2, Returns **1** along with the first location in the string s2,
ii) if s1 is **equal** to s2 , returns **0** and
iii) **otherwise**, returns **-1**.
Write a C program that invokes ***MatchAny***(s1,s2) for different input strings and output both the strings s1 & s2 with the return value in tabular form. **Note: Do not use the standard library functions**

Objective:

To develop a matchany(a,b) function in C that takes two string parameters and to understand how the string manipulations are done.

Algorithm:

```
Main() funtion

Step1: [Read strings a and b]
          Read a,b

Step 2: [call function matchany(a,b)]
          found= matchany(a,b)

Step 3: if found==-1
                    print"no match found"
          else
                    print the character and the position
          end if
Step 4: [Get the choice from the user to continue]
          read ch

Step 5: if ch==1
                    goto step1
          else
                    goto step6
          end if

Step 6: stop

Match(a,b) function
Step 1: for i=0 to end of string a in step 1
            for j=0 to end of string b in step 1
                    if a[i]=b[j]
                              return i
                    end if
            end for
          end for

Step 2: return -1
```

Problem Statement 11 :

Draw the flowchart and write a *recursive* C function to find the factorial of a number, ***n*!**, defined by *fact*(*n*)=1, if *n*=0. Otherwise *fact*(*n*)=*n***fact*(*n*-1). Using this function, write a C program to compute the binomial coefficient $_nC_r$. Tabulate the results for different values of ***n*** and ***r*** with suitable messages.

Objective:

To learn the implementation of recursive functions and find factorial of a number.

Algorithm

```
Step1. [Read integer n]


Step 2.[check if the number is negative]
        if (n < 0)
                output Negative integers are not allowed
                return 0;

Step 3.[call the function factorial]
                 f = factorial(n);

Step 4.Output factorial of n i.e f

Step 5.Stop


 factorial(n)

  Step 1.if (n = 0)
        return 1;

 Step 2.[call function recursively till base condition]
        return(n * factorial(n-1));
```

Problem Statement 12 :

Given two text documentary files “Ramayana.in” and “Mahabharatha.in”. Write a C program to create a new file “Karnataka.in” that appends the content of file “Ramayana.in” to the file “Mahabharatha.in”. Display the contents of output file “Karnataka.in” on to screen. Also find number of words and newlines in the output file.

Objective:

To understand and use files in programming.

Algorithm

Step1. [Read 2 existing file name]

Step 2. Open 2 files f1 and f2 in read mode.

Step 3.open a new file f3 in write mode.

Step 4.copy content of f1 in f3.

Step 5.copy content of f2 in f3.

Step 6. Close all 3 files

Problem Statement 13 :

Write a C program to maintain a record of **“n”** student details using an array of structures with four fields (Roll number, Name, Marks, and Grade). Each field is of an appropriate data type. Print the marks of the student given student name as input.

Objective:

To Implement and use user defined type structures.

Algorithm:

```
Step 1. [declare a structure student]
        struct student
        {
                char name[30];
                int rollno;
                int marks;
                char grade;
        }st;

Step 2.Enter number of students
Step 3. for loop to read the names and roll numbers
        for(i=0 to n-1)
        Input  data as
        st[i].rollno
        st[i].name
        st[i].marks
        st[i].grade
step 4.input student’s name whose marks has to be displayed as
        stname

Step 5.search if the student exists
        for(i=0 to n-1)
                if(strcmp(stname,st[i].name))
                display st[i].marks
        end  for
step 6. If student not found
        display message student not found

step 7.Stop
```

Problem Statement 14:

14. a. Write a C program using pointers to compute the sum of all elements stored in an array A[n]. Where n is the number of elements.

Objective:

To understand the use of pointers in arrays.

Algorithm:

```
Step 1. Input number of elements as n

Step 2.[input n integers in array A]
        For(i=0 to  n-1)
                Input to a[i]
        End for

Step 3.assign address of array A to a integer pointer
        Ptr=A;

Step 4.[Compute sum]
        For(i=0 to n-1]
                Sum+=*ptr
                Ptr++
        End for

Step 5. Output  sum

Step 6.Stop
```

b. Write a C program to find sum of n elements entered by the user. Demonstrate the program using functions malloc() to allocate memory dynamically and free() to deallocate.

Objective:

To learn dynamic memory allocation and deallocation in C.

Algorithm:

```
Step 1. Input number of elements as n

Step 2.[dynamically allocate memory to declare a array]
        Ptr=(int *) malloc(n*sizeof(int));

For(i=0 to  n-1)
                Input to a[i]
        End for

Step 3. [input data and Compute sum]
        For(i=0 to n-1]
                Input to ptr+i
                Sum+=*(ptr+i)
        End for

Step 4. Output  sum

Step 5. Free(ptr)

Step 6.Stop
```

Problem Statement 15:

a.Write a C program using pointers to find the median of a list of members.

Objective:

To implement list and pointer to list.

Algorithm:

Step 1.Enter number of students

Step 2. Step 2.[dynamically allocate memory to declare a array]

```
Ptr=(int *) malloc(n*sizeof(student));
```

Step 3. for loop to read the names and roll numbers

```
for (int count = 0 to numStudents; count++)

    "Enter number of books (c + 1) << ": ";
     *(numPtr + count);
```

Step 4.
arraySort(numPtr, numStudents);

```
for (startScan = 0; startScan < (numStudents - 1); startScan++)
{
    minIndex = startScan;
    minElem = *(array + startScan);
    for (int index = startScan + 1; index < numStudents; index++)
    {
        if ((*(array + index)) < minElem)
        {
            minElem = *(array + index);
            minIndex = index;
        }
    }
    *(array + minIndex) = *(array + startScan);
    *(array + startScan) = minElem;
}
```

```
  }
median = findMedian(numPtr, numStudents
  double findMedian(int numPtr[], int numStudents)
  {
      double median = 0;

      if(numStudents % 2 == 0)
      {
          median = (*(numPtr  + (numStudents / 2)) - 1) + *(numPtr  + (numStudents / 2));
          median /= 2;
      }
      Else
      {
          median = *(numPtr  + (numStudents / 2));
      }
      return median;
  }
```