Part B: C Programming

Record Page right side: Aim

Program

Output

Result

Record Page left side

Algorithm

1. Program to find the sum of digits and reverse of a number.

```
#include<stdio.h>
#include<conio.h>
void main( )
{
clrscr( );
int num,sum=0,rev=0,d;
printf("Enter the number: ");
scanf("%d",&num);
while(num){
d=num%10;
num=num/10;
sum=sum+d;
rev=rev*10+d;
}
printf("Sumof digits = %d",sum);
printf("\nReverse of the number = %d",rev);
getch( );
}
```

Algorithm

Step .1: start

Step . 2:declare required variables and initialize sum=0 and rev=0

Step . 3: get the value for variable “num”

Step . 4: using while loop, perform step 5 until condition become false

Step . 5: calculate : d=num%10

Num=num/10

Sum=sum+d

Rev=rev*10+d

Step . 6: print the value of sum and reverse

Step . 7:stop

Output

Enter the number: 52

Sumof digits = 7

Reverse of the number = 25

2. Find first n Fibonacci numbers

```
#include<stdio.h>
#include<conio.h>
void main()
{
   int n, first = 0, second = 1, next, c;
   clrscr();
   printf("Enter the number of terms\n");
   scanf("%d",&n);
   printf("First %d terms of Fibonacci series are :-\n",n);

   for ( c = 0 ; c < n ; c++ )
   {
      if ( c <= 1 )
         next = c;
      else
      {
         next = first + second;
         first = second;
         second = next;
      }
      printf("%d\n",next);
   }
    getch();
}
```

Algorithm

Step1: Start the program

Step 2: Declare required variables

Step 3 : Get the limit n

Step 4 : Using for loop perform step 5 to until condition become false

Step 5: Find Fibonacci numbers by

next = first + second;

first = second;

second = next;

Step 6: Print n Fibonacci numbers

Step 7: Stop

Output

```
Enter the number of terms
5
First 5 terms of Fibonacci series are :-
0
1
1
2
3
```

3. Create a pyramid using '*'

```
#include <stdio.h>
int main()
{
 int i, space, rows, k=0;
 printf("Enter number of rows: ");
 scanf("%d",&rows);
 for(i=1; i<=rows; ++i, k=0)
 {
 for(space=1; space<=rows-i; ++space)
 {
 printf(" ");
 }
 while(k != 2*i-1)
 {
 printf("* ");
 ++k;
 }
 printf("\n");
 }
 return 0;
}
```

Algorithm

Step 1: start

Step 2; declare required variables and initialize k=0

Step 3: get number of rows ,rows

Step 4: using for loop "i" with initial value zero and condition i<=rows perform step 5 to 8

Step 5: using another for loop "space" with initial value 1 and condition space<=rows-I,

perform step 6

Step 6: perform print function printf(" ")

Step 7: using while loop with condition k!=2*i-1,print "*" and increment k value

Step 8: print next line of pyramid by printf("\n")

Step 9: stop

Output

```
Enter number of rows: 5

      *
     * * *
    * * * * *
   * * * * * * *
  * * * * * * * * *
```

4. Find the number of words in a sentence.

```
#include <stdio.h>
#include <string.h>
void main()
{
 char s[200];
 int count = 0, i;
 printf("enter the string\n");
 scanf("%[^\n]s", s);
 for (i = 0;s[i] != '\0';i++)
 {
 if (s[i] == ' ')
 count++;
 }
 printf("number of words in given string are: %d\n", count + 1);
}
```

Algorithm

Step 1: start

Step 2: declare a char array s[200],and other required variables

Step 3: get the string value “s”

Step 4: using for loop with initial value zero and condition s[i]!=’\0’, perform step 5

Step 5: check condition s[i]==” ” ,if true, perform count++

Step 6: print totals words in given string

Step 7: stop

Output

```
enter the string:
welcome to yims

number of words in given string are: 3
```

5. Check whether a number is prime or not

```
#include<stdio.h>
#include<conio.h>
void main( )
{
clrscr();
int n,i,f=0;
printf("Enter the number: ");
scanf("%d",&n);
for(i=2;i<n;i++)
{
  if(n%i==0)
   {
     f=1;
     break;
   }
}

if(f==0)
   printf("The given number is prime");
else
   printf("The given number is not prime");
getch();
}
```

Algorithm

Step 1: Start

Step 2: Read number n

Step 3: Set f=0

Step 4: For i=2 to n-1

Step 5: If n mod 1=0 then

Step 6: Set f=1 and break

Step 7: Loop

Step 8: If f=0 then

print 'The given number is prime'

else

print 'The given number is not prime'

Step 9: Stop

Output

```
Enter the number: 2
The given number is prime
Enter the number: 4
The given number is not prime
```

6. Perform matrix transpose

```
#include  <stdio.h>
#include<conio.h>
#define ROW 2
#define COL 2
void matrixInput(int mat[][COL]);
void matrixPrint(int mat[][COL]);
void matrixTranspose(int mat[][COL]);
void main()
{
int mat1[ROW][COL];
int mat2[ROW][COL];
int product[ROW][COL],add[ROW][COL],sub[ROW][COL];
clrscr();
printf("Enter elements in matrix of size %dx%d\n", ROW, COL);
matrixInput(mat1);
printf("\n matrix before transpose \n");
matrixPrint(mat1);
matrixTranspose(mat1);
getch();
}
void matrixInput(int mat[][COL])
{
int row, col;
for (row = 0; row < ROW; row++)
{
for (col = 0; col < COL; col++)
{
scanf("%d", (*(mat + row) + col));
}
}
}
void matrixPrint(int mat[][COL])
{
int row, col;
for (row = 0; row < ROW; row++)
```

```
{
for (col = 0; col < COL; col++)
{
printf("%d ", *(*(mat + row) + col));
}
printf("\n");
}
}
void matrixTranspose(int mat[][COL])
{
int row, col,trans[ROW][COL];
for (row = 0; row < ROW;row++)
{
for (col = 0; col < COL; col++)
{
*(*(trans + col) + row) = *(*(mat + row) + col);
}
}
printf("matrix after transpose\n");
for (row = 0; row < ROW; row++)
{
for (col = 0; col < COL; col++)
{
printf("%d ", *(*(trans + row) + col));
}
printf("\n");
}
}
```

Algorithm

Step 1: start

Step 2: declare functions like,matrixinput(),matrixprint(),matrixtranspose with required attributes and variables

Step 3: declare 2D matrix mat1[][]

Step 4: get values to mat1[][]using matrixinput()

Step 5: perform transpose of matrix using matrixtranspose() with arguments mat1 and print the transpose of matrix mat1

Step 9: stop

OUTPUT

```
Enter elements in matrix of size 2x2
1
2
3
4

 matrix before transpose
1 2
3 4
matrix after transpose
1 3
2 4
```

7. Find the sum of the series S = 1 + (½)2 +(1/3)3 +. .....to 0.0001% accuracy.

```
#include   <math.h>
#include   <stdio.h>
double Series(int n)
{
int i;
double sums = 0.0, ser;
for (i = 1; i <= n; ++i) {
ser = 1  /  pow(i,  i);
sums +=  ser;
}
return sums;
}
int main()
{
int n ;
printf("Enter the limit ");
scanf("%d",&n);
double res = Series(n);
printf("\n sum of the series is %f", res);
return 0;
}
```

Algorithm

Step 1: start

Step 2: get the value for “n”

Step 3: call the function “series()” with pass by value “n” and assign the return value to “res”

Step 4: within the function series(), declare required variables and assign sums=0.0

Step 5: perform ser=1/pow(i,i) using for loop “i”

Step 6: calculate sum+=ser

Step 7: return “sums” value to main()

Step 8: stop

Output

Enter the limit 5

Sum of the series is 1.291263

8. Create a pattern with the number N.

```
#include  <stdio.h>
#include<conio.h>
void main()
{
    long n , i=1;
    clrscr();
    printf("Enter number");
    scanf("%ld",&n);
    printf("\n Pattern \n");
    for(i=10;i<n;i*=10);
    for (i=i/10; n>0; i/=10)
    {   printf("%ld \n", n);
          n%=i;
    }
    getch();
}
```

Algorithm

Step 1: start the program

Step 2: get value for variable n

Step 3: using for loop as special case, to calculate i*=10 by implementing special loop

structure

Step 4: using another for loop with initial value i=i/10 and condition n>10 perform step 5

repeatedly by increment statement 1/=10

Step 5: print the value of n and also perform n%=i

Step 6: stop

Output

Enter a number 39174

Pattern: 3 9 1 7 4

9 1 7 4

1 7 4

7 4

4

9. Display the short form of a string. E.g. Computer Science : CS

```
#include<stdio.h>
#include<string.h>
void main()
{
char str[100],*ptr,i,l;
clrscr();
printf("Enter any string\n");
gets(str);
l=strlen(str);
ptr=str;
printf("%c",*(ptr+0));
for(i=1;i<l;i++)
{
if(*(ptr+i-1)==' ')
printf(" %c ",*(ptr+i));
}
getch();
}
```

Algorithm

Step 1: Start the program

Step 2 : Get string Example " Computer science Yuvakshetra"

Step 3 : Find length of the string using strlen() function.

Step 4 : using for loop with limit string length find first letters of the words in given string.

Step 5 : Print abbreviation of string.

Step 6 : Stop

OUTPUT

```
Enter any string
computer science yuvakshetra
c s  y _
```

10. Find the currency denomination of a given amount

```
#include <stdio.h>
#include <conio.h>
#define SIZE  9
void main()
{
    int amount, notes,i;
        int denominations[SIZE] = { 2000, 500, 200, 100, 50, 20, 10, 5, 1 };\
clrscr();
    printf("Enter amount: ");
    scanf("%d", &amount);
    printf("\n");
    for (i=0;i<SIZE;i++)
    {
                notes = amount / denominations[i];
                if (notes)
                {
                    amount = amount % denominations[i];
                    printf("%d * %d = %d \n", notes, denominations[i],
                   notes * denominations[i]);
                }
    }
    getch();

}
```

ALGORITHM

Step 1: Start the program

Step 2: The user-defined function denomination will divide the amount in to 2000, 500,100,50,20,10,5,2,1 rupees notes.

Step 3 : Get the amount

Step 4: Using for loop with limit SIZE find number of notes and denominations using formulas.

Step 5: Print denominations of given number.

Step 6 : Stop

OUTPUT

```
Enter amount: 254321

-3 * 2000 = -6000
-3 * 500 = -1500
-1 * 200 = -200
-1 * 100 = -100
-1 * 20 = -20
-3 * 1 = -3
```

11. Find the Armstrong numbers within a given range.

```
#include  <stdio.h>
#include  <math.h>
int main()
{
 int low, high, i, temp1, temp2, remainder, n = 0, result = 0;
 printf("Enter two numbers(intervals): ");
 scanf("%d %d", &low, &high);
 printf("Armstrong numbers between %d an %d are: ", low, high);
 for(i = low + 1; i < high; ++i)
 {
 temp2 = i;
 temp1 = i;
 while (temp1 != 0)
 {
 temp1 /= 10;
 ++n;
 }
 while (temp2 != 0)
 {
 remainder = temp2 % 10;
 result += pow(remainder, n);
 temp2 /= 10;
 }
if (result == i) {
 printf("%d ", i);
 }
 n = 0;
 result = 0;
 }
 return 0;
}
```

Algorithm

Step 1: start

Step 2: get values for variable “low” and ”high”

Step 3: using for loop with initial value “low” and limit “high” ,perform step 4 to8

Step 4: assign value of “I” to temp2 and temp1

Step 5: using while loop with condition temp1!=0,perform temp/=10 and increment “n” value

Step 6: using while loop with condition temp2!=0 perform

Reaminder=temp%10

Result+=pow(remainder,n)

Temp2/=10

Step 7: if result value equal to one, printvalue of “I”

Step 8: assign n and result value with zero

Step 9: stop

Output

Enter two numbers (intervals):

10

500

Armstrong numbers between 10 an 500 are:

153

370

371

407

12. Find the factorial of a number using recursion

```
#include<stdio.h>
#include<conio.h>
long  int  fact(int  n);
void main() {
    int n;
    clrscr();
    printf("Enter a positive integer: ");
    scanf("%d",&n);
    printf("Factorial of %d = %ld", n, fact(n));
    getch();
}

long int fact(int n)
{
    if (n>=1)
        return n*fact(n-1);
    else
        return 1;
}
```

Algorithm

Step 1: start

Step 2: declare function fact(int n)and required varibles.

Step 3: get positive value for variable “n”

Step 4: call function fact(n); and perform step 5 and print factorial value

Step 5:

```
long int fact(int  n)
{
    if (n>=1)
        return n*fact(n-1);
    else
        return 1;
}
```

Step 6: stop

OUTPUT

```
Enter a positive integer: 5
Factorial of 5 = 120
```

13. Check for palindrome string

```
#include <stdio.h>
#include <string.h>
int main(){
 char string1[20];
 int i, length;
 int flag  =  0;
 printf("Enter a string:");
 scanf("%s", string1);
 length = strlen(string1);
 for(i=0;i < length  ;i++){
 if(string1[i] != string1[length-i-1]){
 flag = 1;
 break;
 }
}
 if (flag) {
 printf("%s is not a palindrome", string1);
 }
 else {
printf("%s is a palindrome", string1);
 }
 return 0;
}
```

Algorithm

Step 1: start

Step 2: declare required variables and get the value for the string “string1”

Step 3: calculate length of string using strlen() and assign the value to variable “length”

Step 4: using for loop with limit “length” perform step 5

Step 5: set flag=1, if the condition string[i].string1[length-i-1] is true

Step 6: if flag value is zero, print not palindrome, else print palindrome

Step 7: stop

Output

```
Enter a string:
malayalam
malayalam is a palindrome
```

14. Check for leap year.

```
#include   <stdio.h>
#include  <conio.h>
void main()
{
 int year;
 printf("Enter a year: ");
scanf("%d",&year);
if(year%4 == 0)
 {
 if( year%100 == 0)
 {
 if ( year%400 == 0)
 printf("%d is a leap year.", year);
 else
 printf("%d is not a leap year.",  year);
 }
 else
 printf("%d is a leap year.", year  );
 }
 else
 printf("%d is not a leap year.", year);
 getch();
}
```

Algorithm

Step 1: start

Step 2: declare and get a value for variable “year”

Step 3: if year%4 == 0, year%100 == 0) and year%400 == 0) is true then
print leap year, else print not a leap year

Step 4: stop

Output

Enter a year: 2020

2020 is a leap year.

15. Write odd and even numbers into separate files.

```
#include<stdio.h>
#include<math.h>
void main()
{
FILE  *all,*even,*odd;
int   number,i,records;
clrscr();
printf("INPUT THE TOTAL NUMBER OF RECORDS THAT U WANT TO ENTER");
scanf("%d",& records);
printf("enter numbers");
all=fopen("ANYNUMBER","w");
for(i=1;i<=records;i++)
{
scanf("%d",&number);
if(number==-1)break;
putw(number,all);
}
fclose(all);
all=fopen("ANYNUMBER","r");
even=fopen("EVENNUMBER","w");
odd=fopen("ODDNUMBER","w");
while((number=getw(all))!=EOF)
{
if(number%2==0)
putw(number,even);
else
putw(number,odd);
}
fclose(all);
fclose(even);
fclose(odd);
even=fopen("EVENNUMBER","r");
odd=fopen("ODDNUMBER","r");
printf("THE EVEN NUMBERS ARE");
while((number=getw(even))!=EOF)
```

```
printf("\n %d",number);
printf("\n THE ODD NUMBERS ARE");
while((number=getw(odd))!=EOF)
printf("\n %d\n",number);
fclose(even);
fclose(odd);
getch();
}
```

Algorithm

Step 1: start

Step 2; declare file pointer “all,”even” and “odd”, also declare required variables

Step 3: get total number of records to variable “number”

Step 4: using fopen(),open file “ANY NUMBER” in write mode and assign value to file pointer “all”

Step 5: using for loop with limit records,get value to variable “number”

Step 6; if value of number ==-1, break

Step 7: write number to file pointer “all” using putw()

Step 8: close all files using fclose()

Step 9: open “ANYNUMBER” “ODDNUMBER”EVENNUMBER” files in read and write mode respectively

Step 10: calculate odd and even number and assign values to filepointer “even” and “odd”

Step 11: close all files

Step 12 : print even and odd numbers from respective file

Step 13: stop

OUTPUT

```
INPUT THE TOTAL NUMBER OF RECORDS THAT U WANT TO ENTER
5
Enter numbers
1
2
3
4
5
THE EVEN NUMBERS ARE
2
4
THE ODD NUMBERS ARE
 1
 3
 5
```

16. Base conversion of numbers

```
#include  <stdio.h>
#include<conio.h>
void main()
{
char base_digits[16] =
{'0', '1', '2', '3', '4', '5', '6', '7',
'8', '9', 'A', 'B', 'C', 'D', 'E', 'F'};
int converted_number[64];
int number_to_convert;
int next_digit, base, index=0;
clrscr();
printf("Decimal to other base conversion program \n");
printf("Enter a decimal number ");
scanf("%d",&number_to_convert);
printf("Enter desired base to convert: ");
scanf("%d",&base);
while (number_to_convert != 0)
{
converted_number[index] = number_to_convert % base;
number_to_convert = number_to_convert /  base;
++index;
}
--index;
printf("\n\nConverted Number from base 10 to base %d= ",base);
for( ; index>=0; index--)
{
printf("%c", base_digits[converted_number[index]]);
}
printf("\n");
getch();
}
```

Algorithm

Step 1: start

Step 2: declare an array base_digits[16] and assign array values to it

Step 3: get a decimal value and base value to the respective variables

Step 4: using while loop with condition number_to_convert !=0, perform conversion calculations

Step 5: decrement index value

Step 6: using for loop print new number (converted to new base)

Step 7: stop

Output

Decimal to other base conversion program

Enter a decimal number 9

Enter desired base to convert: 2

Converted Number from base 10 to base 2= 1001

17. Merge two numeric arrays in sorted order.

```
#include <stdio.h>
void main()
{
 int arr1[100], arr2[100], arr3[200];
 int s1, s2, s3;
 int i, j, k;
 clrscr();
 printf("\n\nMerge two arrays in sorted order.\n");
 printf("------------------------------------------------------------------------\n");
 printf("Input the number of elements to be stored in the first array:");
 scanf("%d",&s1);

printf("Input %d elements in the array :\n",s1);
 for(i=0;i<s1;i++)
 {
 printf("element - %d :  ",i);
 scanf("%d",&arr1[i]);
 }
 printf("Input the number of elements to be stored in the second array :");
 scanf("%d",&s2);
 printf("Input %d elements in the array :\n",s2);
 for(i=0;i<s2;i++)
 {
 printf("element - %d :  ",i);
 scanf("%d",&arr2[i]);
 }
 s3 = s1 + s2;
 for(i=0;i<s1; i++)
 {
 arr3[i] = arr1[i];
 }
 for(j=0;j<s2; j++)
 {
 arr3[i] = arr2[j];
 i++;
```

```
 }
 for(i=0;i<s3; i++)
{
 for(k=0;k<s3-1;k++)

{
if(arr3[k]<=arr3[k+1])
{
 j=arr3[k+1];
 arr3[k+1]=arr3[k];
 arr3[k]=j;
}
 }
}
 printf("\nThe merged array in decending order is :\n");
 for(i=s3; i>0; i--)
 {
printf("%d ", arr3[i]);
 }
printf("\n\n");
getch();
}
```

Algorithm

Step 1: start

Step 2: declare, array variables arr1[],arr2[],arr3[] and other required variables

Step 3: using for loops get un sorted array values to arr1[] and arr2[] respectively

Step 4: perform merging and sorting operations using i and j loops with limits s1 and s2 respectively

Step 5: print sorted array arr3[]

Step 6: stop

OUTPUT

```
Merge two arrays in sorted order.
----------------------------------------------------------

Input the number of elements to be stored in the first array :3

Input 3 elements in the array :
element - 0 : 5
element - 1 : 7
element - 2 : 1
Input the number of elements to be stored in the second array :5

Input 5 elements in the array :
element - 0 :  3
element - 1 : 4
element - 2 :  6
element - 3 :  7
element - 4 : 1

The merged array in decending order is :

1 1 3 4 5 6 7 7
```

18. Fill upper triangle with 1, lower triangle with -1 and diagonal elements with 0.

```
#include<stdio.h>
#include<conio.h>
void main()
{
 int i, j,row=3,col=3;
 clrscr();
 printf("Upper Triangle forms like \n");
 for (i = 0; i < row; i++)
 {
for (j = 0; j < col; j++)
{
 if (i<j || i==j)
{
printf("1");
 }
 else
 printf("0");
 printf(" ");
}
printf("\n");
 }

 printf("Lower Triangle forms like \n");

 for (i = 0; i < row; i++)
 {
for (j = 0; j < col; j++)
{
 if (i>j || i==j)
 {
printf("-1");
 }
 else
 printf("0");
 printf(" ");
```

```
}
printf("\n");
 }
 printf(" Triangle with diagonal elements zero \n");
for (i = 0; i < row; i++)
 {
for (j = 0; j < col; j++)
{
 if (i==j)
 {
printf("0");
 }
 else
 printf("1");
 printf(" ");
}
printf("\n");
 }

 getch();
}
```

Algorithm

Step 1: start

Step 2: declare required variables and also assign row and col value as 3

Step 3: using for loop “i” and “j” with limit row and col , check i<j and i==j, ,if true, print upper triangle value 1,

Step 3: using for loop “i” and “j” with limit row and col , check i>j and i==j, ,if true, print lower triangle value -1,

Step 3: using for loop “i” and “j” with limit row and col , check i==j, ,if true, print diagonal value zero,

Step 4: stop

Output

Upper Triangle formslike

1 1 1

0 1 1

0 0 1

Lower Triangle forms like

-1 0 0

-1 -1 0

-1 -1 -1

Triangle with diagonal elements zero

0 1 1

1 0 1

1 1 0

****