Question 20

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

A PROGRAM THAT INPUT A SERIES OF INTEGERS AND PASSES THEM ONE AT A

TIME TO FUNCVTION EVEN ,WHICH USES THE REINDER OPERATOR TO DETERINE

IF AN INTEGER IS EVEN THE FUNCTION TAKE INTEGER ARGUENTS AND RETURN 1 IF INTEGER IS EVEN,OR RETURN O IF INTEGER IS ODD

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<stdio.h>

#include<conio.h>

int num;

int even(int a)

{

if(num%2==0)

return 1;

if(num%2!=0)

return 0;

}

int main()

{

char choice;

do

{

printf("\tEnter integers to check even or odd\n\t");

scanf("%d",&num);

printf("%d",even(num));

printf("\nYou want to enter more integers\t(Y/N)\n\t");

fflush(stdin);

scanf("%c",&choice);

}while(choice=='Y');

return 0;

getch();

}

Question 21

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

A PROGRAM THAT RETURNS THE SALLEST OF THREE FLOATING POINT NUMBERS

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<stdio.h>

#include<conio.h>

int num;

int smallest(int small)

{

int a ,b,c; // variable declaration

printf("Enter 3 floating point values:\n");

scanf("%f %f %f",&a,&b,&c); // taking user input for floatng point values

int sm= (

(((a<b)&&(a<c))

? (printf("a (%f) is smallest number",a))

:((b<c)&&(b<a))?(printf("b ( %f ) is smallest number",b)) : printf("c (%f) is sallest number",c))

) ;

return sm;

}

int main()

{

int val;

printf("%c",smallest(val));

return 0;

getch();

}

Question 22

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*A program that display perfect numbers between 1 and 1000\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

/\* Function declarations \*/

int checkPerfect(int n1);

void PerfectNumbers(int startLimit, int endLimit);

int main()

{

int startLimit=1, endLimit=1000;

printf("\n\n Function : perfect numbers in a given range :\n");

printf("--------------------------------------------------\n");

printf("\nlowest search limit of perfect numbers is %d : ",startLimit);

printf("\nHighest search limit of perfect numbers is %d : ",endLimit);

printf("\n The perfect numbers between %d to %d are : \n", startLimit, endLimit);

PerfectNumbers(startLimit, endLimit);

printf("\n\n");

return 0;

}

// Checks whether the given number is perfect or not.

int checkPerfect(int n1)

{

int i, sum;

sum = 0;

for(i=1; i<n1; i++)

{

if(n1 % i == 0)

{

sum += i;

}

}

// If sum of proper positive divisors equals to given number

// then the number is perfect number

if(sum == n1)

return 1;

else

return 0;

}

void PerfectNumbers(int startLimit, int endLimit)

{

/\* print perfect numbers from start to end \*/

while(stLimit <= endLimit)

{

if(checkPerfect(startLimit))

{

printf(" %d ", startLimit);

}

startLimit++;

}

}

/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*write a function which determines if a number is prime or not\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<stdio.h>

int check\_prime(int);

main()

{

int n, result;

printf("Enter an integer to check whether it is prime or not.\n");

scanf("%d",&n);

result = check\_prime(n);

if ( result == 1 )

printf("%d is prime.\n", n);

else

printf("%d is not prime.\n", n);

return 0;

}

int check\_prime(int a)

{

int c;

for ( c=2;c<=a-1;c++)

{

if ( a%c == 0 )

return 0;

}

return 1;

}

Question 23

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* C program to list all prime number between an interval using function.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

/\* Function declarations \*/

int isPrime(int num);

void printPrimes(int lowerLimit, int upperLimit);

int main()

{

int lowerLimit=1, upperLimit=50;

// Call function to print all primes between the given range.

printPrimes(lowerLimit, upperLimit);

return 0;

}

Question 22(B)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

A program to print all prime numbers between lower limit and upper limit.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void printPrimes(int lowerLimit, int upperLimit)

{

printf("All prime number between %d to %d are: ", lowerLimit, upperLimit);

while(lowerLimit <= upperLimit)

{

// Print if current number is prime.

if(isPrime(lowerLimit))

{

printf("\n");

printf("\t %d ", lowerLimit);

}

lowerLimit++;

}

}

/\*

\* Check whether a number is prime or not.

\* Returns 1 if the number is prime otherwise 0.

\*/

int isPrime(int num)

{

int i;

for(i=2; i<=num/2; i++)

{

/\*

\* If the number is divisible by any number

\* other than 1 and self then it is not prime

\*/

if(num % i == 0)

{

return 0;

}

}

return 1;

}

Question 24

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Write a program that takes an integer value and returns the number with its digits reversed

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include<conio.h>

#include<stdio.h>

int main()

{

int num;

printf("\tEnter 5 digit number:\n\t");

scanf("%d", &num);

printf("\t---------------------\n");

printf("\tReversed 5 digit number:\n\t");

int num1=num%10;

printf("\t%d",num1);

int num2 =(num/10)%10;

printf("%d",num2);

int num3 =((num/10)/10)%10;

printf("%d",num3);

int num4 =(((num/10)/10)/10)%10;

printf("%d",num4);

int num5 =((((num/10)/10)/10)/10)%10;

printf("%d",num5);

getch();

return 0;

}

Question 25

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Guess Game\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdlib.h>

#include <conio.h>

#include <stdio.h>

int high=1000,low=1;

int n,secnum;

int guess(int k)

{

secnum = rand()%(high-low)+low;

if(n>secnum)

{

printf("\nOops! Guess was too high ");

printf("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\t");

}

else if(n<secnum)

{

printf("\nOops! Guess was too low");

printf("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\t");

}

else if(n == secnum)

{

printf("\n\t\tHurrah! You win");

printf("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\t");

}

}

int main()

{

char ch;

printf("WELLCOME TO THE GUESS GAME %c",1);

printf("\n \tI have a number between 1 to 1000\n \t can you guess my number\n");

printf("\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\t");

printf("\nPlease type your first guess\n \t===");

scanf("%d",&n);

printf("\nExcellent you make a guess");

printf("%d",guess (n));

printf("\nWould you like to try again?\n\t(y/n)\_\_\_\_\_\_\_\_");

fflush(stdin);

scanf("%c",&ch);

while(ch=='y')

{

do{

printf("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\t");

printf("\n Make a guess===");

scanf("%d",&n);

printf("%d",guess (n));

printf("\nWould you like to try again?\n\t(y/n)\_\_\_\_\_\_\_\_\_\n");

fflush(stdin);

scanf("%c",&ch);

} while(ch=='y');

}

printf("\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\t");

printf("\nSecret Number was %d",secnum);

getch();

return 0;

}

Question 26(a)

//Fibonacci series program in C

#include <stdio.h>

int n=50, first = 0, second = 1, next, c;

int fab()

{

for (c = 0; c < n; c++)

{

if (c <= 1)

next = c;

else

{ next = first + second;

first = second;

second = next;

}

printf("%d\n", next);

}

}

int main()

{

printf("First %d terms of Fibonacci series are:\n", n);

printf("%d",fab());

return 0;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Program to print Fibonacci Sequence using recursion

\*

\* Enter terms: 10

\* 0 1 1 2 3 5 8 13 21 34

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

Question 26(b)

#include<stdio.h>

int num=0 , terms=50;

int fibonacci();

int series()

{

if( num < terms)

printf("%d ", fibonacci());

}

int main(void)

{

printf(" 50 terms of Fabonaccii series: ");

printf("%d",series);

return 0; // return 0 to operating system

}

int fibonacci(num)

{

if(num == 0 || num == 1)

{

return num;

}

else

{

// recursive call

return fibonacci(num-1) + fibonacci(num-2);

}

}

Question 28

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* TEST PALINDROME \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <string.h>

#include <conio.h>

#include<stdio.h>

int checkpalindrome(char \*s)

{

static int i,c=0,n=strlen(s);

if(i<n/2)

{

if(s[i]==s[n-i-1])

c++;

i++;

checkpalindrome(s);

}

else

{

if(c==i)

return 1;

else

return 0;

}

}

int main()

{

char s[1000];

printf("Enter the string: \n\t");

gets(s);

if(checkpalindrome(s)==1)

printf("string is palindrome");

else

printf("string is not palindrome");

}

Question 29

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* print array\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <stdio.h>

#include<stdlib.h>

#define MAX\_SIZE 100

/\* Function declaration \*/

int start, len;

int printArray(int arr[], int, int);

int main()

{

int arr[MAX\_SIZE];

int N, i;

/\* Input size and elements in array \*/

printf("Enter size of the array: ");

scanf("%d", &N);

/\* Prints array recursively \*/

if(arr[i]==0)

{

exit(0);

}

if(arr[i]!=0)

{

printf("Enter elements in the array: ");

for(i=0; i<N; i++)

{

scanf("%d", &arr[i]);

}

printf("Elements in the array: ");

printArray(arr, 0, N);

/\* Prints the current array element \*/

printf("%d, ", arr[start]);

/\* Recursively call printArray to print next element in the array \*/

printArray(arr, start + 1, len);

}

return 0;

}

/\*\*

\* Prints an array recursively within a given range.

\*/

int printArray(int arr[], int start,int len)

{

if(start==0)

return 0;

/\* Recursion base condition \*/

else if(start >= len)

return 1;

}