**Problem 1: Celsius to Fahrenheit**

Given a temperature in Celsius C. You have to convert it in Fahrenheit

ANSWER:

#include <stdio.h>

*int* main()

{

*float* temp,f;

printf("enter the temperature in celcius:");

scanf("%f",&temp);

f=(temp\*9/5)+temp;

printf("temperature in fahrenheit:%0.2f",f);

}

**Problem 2: Reverse digits of a number**Given an integer N, reverse the digits of N. Rmember that you need to make changes to the same number.

ANSWER:

#include <stdio.h>

*int* main()

{

*int* num,rev\_num=0;

printf("enter the number:");

scanf("%d",&num);

printf("before reversing:%d\n",num);

while (num>0)

{

    rev\_num=rev\_num\*10+num%10;

    num=num/10;

}

printf("after reversing:%d",rev\_num);

}

**Problem 3: Change time format**

Given a time T in 12hr clock format [HH:MM AM/PM], change it to 24hrclock format [HHMM hrs]. You can choose input type as per your convenience.

ANSWER:

#include <stdio.h>

int main()

{

int hour,minutes,c;

char ch;

printf("enter the hour:");

scanf("%d",&hour);

printf("enter the minutes:");

scanf("%d",&minutes);

while ((getchar()) != '\n') ;

printf("is it morning? enter Y/N:");

scanf("%c",&ch);

if(ch=='Y'||ch=='y')

{

    printf("time in 12 hr format= %02d:%02dAM\n",hour,minutes);

    if(hour<12)

    {

        printf("time in 24 hr format= %02d%02d HRS",hour,minutes);

    }

    else

    {

       hour=(hour+12);

       printf("time in 24 hr format= %02d%02d HRS",hour,minutes);

    }

}

else if (ch =='N'||ch=='n')

{

printf("time in 12 hr format= %02d:%02dPM\n",hour,minutes);

hour=(hour+12);

printf("time in 24 hr format= %02d%02d HRS",hour,minutes);

}

return 0;

}

**Problem 4: Get rid of boring Maths?**

Complete the function Squareroot(int N), whch takes N as argumentand returns the square root of N.

ANSWER:

#include <stdio.h>

#include <math.h>

*double* Squareroot(*int* *N*)

{

    return sqrt(*N*);

}

*int* main()

{

*int* N;

printf("enter the number whose square root is to be calculated: ");

scanf("%d",&N);

*double* a = Squareroot(N);

printf("squareroot=%lf",a);

return 0;

}

**Problem 7: Happy Hours**

Tanay always tells time in minutes, but the students find it difficult to comprehend. Design a function that converts [minutes] into [hour sand minutes] which makes time reading more comprehensive.

ANSWER:

#include <stdio.h>

int min;

int happyHours(int min)

{

    int hour= min/60;

    int minute=min%60;

    printf("time in hours and minute=%d:%02d",hour,minute);

}

int main()

{

    printf("enter the time in minutes:");

    scanf("%d",&min);

    happyHours(min);

}

**Problem 12: Pattern**

Print the following pattern using loop:

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

ANSWER:

#include <stdio.h>

int main() {

 for (int i = 1; i <= 5; i++) {

        for (int k = i; k <= 5; k++) {

            printf("  ");

        }

        for (int j = 1; j <= i; j++) {

            printf("\* ");

        }

        printf("\n");

    }

    return 0;

}

**Problem 13: Nth Fibonacci**

Take N as input. Print Nth Fibonacci Number, given that the first two numbers in the Fibonacci Series are 0 and 1.

ANSWER):

#include <stdio.h>

int fibonacci(int n)

{

    int f1=0,f2=1,f3;

    if(n==0)

    {

        return f1;

    }

    for(int i=2;i<=n;i++)

    {

    f3=f1+f2;

    f1=f2;

    f2=f3;

    }

    return f2;

}

int main()

{

    int num;

    printf("enter a number for nth fibonacci: ");

    scanf("%d",&num);

    printf("the result is %d",fibonacci(num));

}

**Problem 14: N factorial using Recursion**

For a given number N, calculate and return N factorial, use recursion.

ANSWER):

#include <stdio.h>

int factorial(int n)

{

    if (n==0||n==1)

    {

      return 1;

    }

    int small\_calc=factorial(n-1);

    int result= small\_calc\*n;

    return result;

}

int main()

{

    int number;

    printf("enter the number for its factorial:");

    scanf("%d",&number);

    printf("factorial=%d",factorial(number));

}

**Problem 17: FizzBuzz**

Given an integer n, return a string array answer (1-indexed) where:

answer[i] == "FizzBuzz" if i is divisible by 3 and 5.

answer[i] == "Fizz" if i is divisible by 3.

answer[i] == "Buzz" if i is divisible by 5.

answer[i] == i (as a string) if none of the above conditions are true.

ANSWER):

#include <stdio.h>

int fizzBuzz(int n)

{

    for(int i=1;i<=n;i++)

    {

        if(i%3==0&&i%3==0)

        {

            printf("FizzBuzz\n");

        }

        else if(i%3==0)

        {

            printf("Fizz\n");

        }

        else if (i%5==0)

        {

            printf("Buzz\n");

        }

        else

        {

            printf("%d\n",i);

        }

    }

}

int main()

{

    int num;

    printf("enter the number: ");

    scanf("%d",&num);

    fizzBuzz(num);

    return 0;

}

**Problem 18: Max of an array**

Given an integer array **nums**, return the largest element present in thearray.

**Sample Input:** [5, 0, -1, 3, 6, 9, 2, 7]

**Sample Output:** 9

ANSWER):

#include <stdio.h>

int largestNum(int arr[], int n)

{

    int i;

    int max = arr[0];

    for (i = 1; i < n; i++)

        if (arr[i] > max)

            max = arr[i];

    return max;

}

int main()

{

    int arr[] = {5, 0, -1, 3, 6, 9, 2, 7 };

    int n = sizeof(arr) / sizeof(arr[0]);

    printf("Largest in given array is %d", largestNum(arr, n));

    return 0;

}

**Problem 19: Find Duplicates**

Given an array of integers **nums** containing n + 1 integers where eachinteger is in the range [1, n] inclusive.

There is only one repeated number in nums, return this repeatednumber.

You must solve the problem without modifying the array nums anduses only constant extra space.

**Example 1:**

Input: nums = [1,3,4,2,2]

Output: 2

ANSWER):

#include <stdio.h>

int main()

{

    int a[] = {1,3,4,2,2};

    int length = sizeof(a)/sizeof(a[0]);

    printf("Duplicate element in given array: \n");

    for(int i = 0; i < length; i++) {

        for(int j = i + 1; j < length; j++) {

            if(a[i] == a[j])

                printf("%d\n", a[j]);

        }

    }

    return 0;

}

**Problem 20: Sort the array**

Given an integer array **nums**, sort it in descending order (from largestto smallest element) and return the sorted array.

ANSWER):

#include <stdio.h>

int main()

{

    int a[100],n,i,j;

    printf("Array size: ");

        scanf("%d",&n);

        printf("Elements: ");

      for(i=0;i<n;i++)

    {

        scanf("%d",&a[i]);

    }

    for (int i = 0; i < n; i++)

    {

        for (int j = 0; j < n; j++)

        {

            if (a[j] < a[i])

            {

                int tmp = a[i];

                a[i] = a[j];

                a[j] = tmp;

            }

        }

    }

    printf("\n\nDescending : ");

    for (int i = 0; i < n; i++)

        printf(" %d ", a[i]);

    return 0;

}