SAVEETHA SCHOOL OF ENGINEERING

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES

DEPARTMENT OF DATA SCIENCE

COURSE CODE :  DSA01

COURSE NAME : OBJECT ORIENTED PROGRAMMING WITH C++

C++ PRACTICE PROGRAMS DAY 5

QUESTION 1

1. *Problem Statement:* The packing department of a television vision set manufacturer has to prepare a requisition note listing the number of different boxes required for the different TV models that it has received from the production department. The list prepared has to be forwarded to the stores department so that the required boxes are issued to the packing department. The category and the number of boxes required for each type of TV model is given as follows: Model types are TV-LCD 17,22,26,32,37 and box types are 1,2,3,4,2 respectively.

Analysis:

Output

prepare a requisition note include TV model, box types, number of different boxes required

Input: number of different boxes required for the different TV models

OUTPUT

#include<iostream>

using namespace std;

int main()

{

int tv17box,tv22box,tv26box,tv32box,tv37box;

int tv17num,tv22num,tv26num,tv32num,tv37num;

tv17box=1;

tv22box=2;

tv26box=3;

tv32box=4;

tv37box=5;

cout<<" enter number of tv 17 box model :";

cin>>tv17num;

cout<<" enter number of tv 22 box model :";

cin>>tv22num;

cout<<" enter number of tv 26 box model :";

cin>>tv26num;

cout<<" enter number of tv 32 box model :";

cin>>tv32num;

cout<<" enter number of tv 37 box model :";

cin>>tv37num;

cout<< "\n requesition note ";

cout<<" ============================================ ";

cout<<" \n TV MODEL | BOX TYPE | NUMBERS REQUIRED ";

cout<<"============================================= ";

cout<<"\n tv model of 17 "<<tv17box<<tv17num;

cout<<" \n tv model of 22 "<<tv22box<<tv22num;

cout<<" \n tv model of 26 "<<tv26box<<tv26num;

cout<<" \n tv model of 32 "<<tv32box<<tv32num;

cout<<" \n tv model of 37 "<<tv37box<<tv37num;

return 0;

}

QUESTION 2:

. *Problem Statement* : While purchasing certain items, a discount of 10% is offered if the quantity purchased is more than 1000. If quantity and price per item are input through the keyboard, write a program to calculate the total expenses.

3.Write a source code for banking system with account creation with minimum 500 INR, deposit, withdraw, and balance checking features for 20 holders. Here 500 INR minimum balance should be maintain by all account holders.

OUTPUT

#include <iostream>

using namespace std;

int main()

{

int rows, num = 1;

cout<< "Enter number of rows: ";

cin>>rows;

for(int i = 1; i<= rows; i++)

{

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

{

cout<<num\*num\*num<< " ";

num++;

}

cout<<endl;

}

return 0;

}

QUESTION 3:

*3.Problem Statement* :A Travels company insures its drivers in the following cases:

Driver is married.

Driver is an unmarried male above 30 years of age.

Driver is an unmarried female above 25 years of age.

In all other cases the driver is not insured. If the marital status, gender and age of the driver are the inputs, write a program to determine whether the driver is to be insured or not.

OUTPUT

#include <iostream>

#include <string>

#include <math.h>

#include <stdio.h>

using namespace std;

int main ()

{

int m,s,p;

{

cout<<"-If the driver is married:-\n";

cout<< "1. yes\n";

cout<< "2. no\n";

cin>>m;

while (m==1)

{

cout<<"-You are insured-\n";

return 0;

}

}

{

cout<<"-Enter sex-:\n";

cout<< "1. male\n";

cout<< "2. female\n";

cin>>s;

cout<<"-Enter your age-:\n";

cin>>p;

}

if ((s==1, p>39) and (s==2, p>25))

cout<<"-You are insured-\n";

else

cout<<"-You are not insured-\n";

return 0;

}

QUESTION 4:

. *Problem Statement* :You are playing an online game. In the game, a list of N numbers is given. The player has to arrange the numbers so that all the odd numbers of the list come after the even numbers. Write an algorithm to arrange the given list such that all the odd numbers of the list come after the even numbers.

Input

The first line of the input consists of an integer num, representing the size of the list(N). The second line of the input consists of N space-separated integers representing the values of the list

Output

Print N space-separated integers such that all the odd numbers of the list come after the even numbers

OUTPUT

#include <iostream>

using namespace std;

void swap(int \*a, int \*b);

void segregateEvenOdd(int arr[], int size)

{

int left = 0, right = size-1;

while (left < right)

{

while (arr[left] % 2 == 0 && left < right)

left++;

while (arr[right] % 2 == 1 && left < right)

right--;

if (left < right)

{

swap(&arr[left], &arr[right]);

left++;

right--;

}

}

}

void swap(int \*a, int \*b)

{

int temp = \*a;

\*a = \*b;

\*b = temp;

}

int main()

{

int arr[] = {12, 34, 45, 9, 8, 90, 3};

int arr\_size = sizeof(arr)/sizeof(arr[0]);

int i = 0;

segregateEvenOdd(arr, arr\_size);

cout<<"Array after segregation ";

for (i = 0; i<arr\_size; i++)

cout<<arr[i] << " ";

return 0;

}

QUESTION 5:

*Problem Statement :*

Write a program to print all the locations at which a particular element

(taken as input) is found in a list and also print the total number of times it occurs

in the list. The location starts from 1.

For example if there are 4 elements in the array

5

6

5

7

If the element to search is 5 then the output will be

5 is present at location 1

5 is present at location 3

5 is present 2 times in the array

OUTPUT

#include<iostream>

using namespace std;

int main()

{

int i,j,k,n,a[30];

cout<<"How many elements?";

cin>>n;

cout<<"\nEnter elements of array\n"<<endl;

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

cin>>a[i];

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

for(j=i+1;j<n;)

{

if(a[i]==a[j])

{

for(k=j;k<n-1;++k)

a[k]=a[k+1];

--n;

}

else

++j;

}

cout<<"\n";

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

cout<<a[i]<<" ";

return 0;

}

QUESTION 6:

*9.Problem Statement :*

Write a program to print the below pattern

1

8 27

64 125 216

343 512 729 1000

OUTPUT

#include <iostream>

using namespace std;

int main()

{

int rows, num = 1;

cout<< "Enter number of rows: ";

cin>>rows;

for(int i = 1; i<= rows; i++)

{

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

{

cout<<num\*num\*num<< " ";

num++;

}

cout<<endl;

}

return 0;

}

QUESTION 7:

*10.Problem Statement :* A child is running up a staircase with n steps and can hop either 1 step, 2 steps, or 3 steps at a time. Implement a method to count how many possible ways the child can run up the stairs.  
Examples:

Input : 4

Output : 7

OUTPUT

#include<iostream>

using namespace std;

int count\_Ways(int n)

{

if(n<0)

{

return 0;

}

else if(n==0)

{

return 1;

}

else

{

return count\_Ways(n-1) +count\_Ways(n-2) + count\_Ways(n-3);

}

}

int main()

{

int n;

cout<<"Enter number of stairs: ";

cin>>n;

cout<<"There are "<<count\_Ways(n)<<" possible ways the child can run up thestairs."<<endl;

return 0;

}

QUESTION 8:

12.Write a program to print number of factors and to print nth factor of the given number.

Sample Input:

Given Number: 100

N = 4

Sample Output:

Number of factors = 9

4th factor of 100 = 5

Test Cases:

Given Number = 512 , N = 6

Given Number = 343 , N = 7

OUTPUT

#include<iostream>

#include<string.h>

using namespace std;

int main(){

int n,num;

cout<<"Enter Number (between 1 and 100,000,00) : ";

cin>>num;

cout<<"Enter nth Number";

cin>>n;

int count=0,temp=0;

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

if(num%i==0){

count++;

if(count==n){

temp=i;

}

}

}

cout<<"Number of factors ="<<count;

cout<<"nth factor "<<n,num,temp;

}

QUESTION 9:

13.Write a Program to Remove the Duplicate Items from a array.

Sample Input:

Enter the number of elements in array:7

Enter element1:10

Enter element2:20

Enter element3:20

Enter element4:30

Enter element5:40

Enter element6:40

Enter element7:50

Sample Output:

Non-duplicate items:

[10, 20, 30, 40, 50]

OUTPUT

#include<iostream>

using namespace std;

int main ()

{

int A[10], B[10], n, i, j, k = 0;

cout<< "Enter size of array : ";

cin>>n;

cout<< "Enter elements of array : ";

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

cin>> A[i];

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

{

for (j = 0; j < k; j++)

{

if (A[i] == B[j])

break;

}

if (j == k)

{

B[k] = A[i];

k++;

}

}

cout<< "Repeated elements after deletion : ";

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

cout<< B[i] << " ";

return 0;

}

QUESTION 10:

14.Program to find whether the given number is Armstrong number or not

Sample Input:

Enter number : 153

Sample Output:

Given number is Armstrong number

Test cases:

370

1

371

145678

0.21345

OUTPUT

#include <iostream>

using namespace std;

int main() {

int num, originalNum, remainder, result = 0;

cout<< "Enter a three-digit integer: ";

cin>>num;

originalNum = num;

while (originalNum != 0) {

remainder = originalNum % 10;

result += remainder \* remainder \* remainder;

originalNum /= 10;

}

if (result == num)

cout<<num<< " is an Armstrong number.";

else

cout<<num<< " is not an Armstrong number.";

return 0;

}