1. **A) Write a C++ program to perform basic arithmetic operations.**

/\*

\* C++ program to demonstrate use of arithmetic operators

\*/

#include<iostream>

using namespace std;

int main()

{

int Num1, Num2;

cout << "Enter two numbers to be operated with arithmetic operators: ";

cin >> Num1 >> Num2;

cout << endl;

cout << "Num1 + Num2 = " << Num1 + Num2 << endl;

cout << "Num1 \* Num2 = " << Num1 \* Num2 << endl;

cout << "Num1 - Num2 = " << Num1 - Num2 << endl;

if (Num2 != 0)

cout << "Num1 / Num2 = " << Num1 / Num2 << endl;

else

cout << "Num2 is not non-zero. Division is not defined. " << endl;

return 0;

}

**Output:**

Enter two numbers to be operated with arithmetic operators: 2 4

Num1 + Num2 = 6

Num1 \* Num2 = 8

Num1 - Num2 = -2

Num1 / Num2 = 0

1.b)

|  |
| --- |
| **Given an integer variable “days” that contains a number of days, write a C++ program that displays**  **i) the number of whole weeks corresponding to days. For example, if days contains 23, then the number of whole weeks is 3.**  **ii) the number of days remaining after taking the whole weeks out of the value in days. For example, if days contains 23, then the number of days remaining after 3 whole weeks is 2.** |

#include<iostream>

using namespace std;

int main()

{

int d,w;

cout << " Enter the number of days" <<endl;

cin>>d;

w=d/7;

d=d%7;

cout << "Number of weeks =" <<w<<"\tNumber of days ="<<d<<endl;

return 0;

}

**Output:**

Enter the number of days

23

Number of weeks 3 Number of days 2

1. A) Write a C++ program that determines the median of three input numbers. The median is the middle number when the three numbers are arranged in order by size. However, the user can input the values in any order.

#include <iostream>

using namespace std;

// Function to find the middle of three numbers

int middleOfThree(int a, int b, int c)

{

// Compare each three number to find middle

// number. Enter only if a > b

if (a > b)

{

if (b > c)

return b;

else if (a > c)

return c;

else

return a;

}

else

{

// Decided a is not greater than b.

if (a > c)

return a;

else if (b > c)

return c;

else

return b;

}

}

// Driver Code

int main()

{

int a,b,c;

cout<<"enter a, b, c values"<<endl;

cin>>a>>b>>c;

cout << middleOfThree(a, b, c);

return 0;

}

Output:

enter a, b, c values

20 25 15

20

2. b) Write an interactive C++ program which takes input of a series of n temperatures entered by the user, calculates the average temperature.

#include <iostream>

using namespace std;

int main()

{

int n, i;

float num[1000], sum=0.0, average;

cout << "Enter the numbers of days: ";

cin >> n;

while (n > 1000 || n <= 0)

{

cout << "Error! number should in range of (1 to 1000)." << endl;

cout << "Enter the number again: ";

cin >> n;

}

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

{

cout << "day"<<i + 1 << ". Enter temparature: ";

cin >> num[i];

sum += num[i];

}

average = sum / n;

cout << " tempareture Average = " << average;

return 0;

}

Output:

Enter the numbers of days: 3

day1. Enter temparature: -12

day2. Enter temparature: 14

day3. Enter temparature: 2

tempareture Average = 1.33333

|  |
| --- |
| 3 A) ROT13 (rotate by 13 places) is a simple letter substitution cipher that is an instance of a Caesar cipher developed in ancient Rome and used by Julius Caesar who used it in his private correspondence. ROT13 replaces a letter with the letter 13 letters after it in the alphabet. The following example demonstrates the translation in ROT13:  A → N B → O C→P D → Q  Thus, the translation of the word JULIUS using ROT13 would be WHYVHF.  Write a C++ program that inputs a sentence and outputs the corresponding translated sentence using the ROT13 encoding scheme above. |

#include<iostream>

#include<string>

using namespace std;

int main()

{

int i=0,len;

string str;

cout<<"Enter input:";

getline(cin,str); //Reading full line

len=str.length(); //Finding length of the line

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

{

if(str[i]>=65 && str[i]<=90) //Checking for Capital alphabets

str[i]= 65 + ((str[i]-65)+13)%26 ; //Rotation by 13(str[i]+13)

else if(str[i]>=97 && str[i]<=122) //Checking for Small alphabets

str[i]= 97 + ((str[i]-97)+13)%26 ; //Rotation by 13(str[i]+13)

}

cout<<str;

}

/\*

c

---

str[i] 99

str[i]-97 2

(str[i]-97)+13 15

(str[i]-97)+13)%26 15

97+(str[i]-97)+13)%26 112 (p)

n

---

str[i] 110

str[i]-97 13

(str[i]-97)+13 26

(str[i]-97)+13)%26 0

97+(str[i]-97)+13)%26 97 (a)

\*/

Output:

Enter input: JULIUS

WHYVHF

|  |
| --- |
| b) Write a C++ program that displays a checkerboard pattern made of stars and blanks, as shown below. A checkerboard is of size eight squares by eight squares. |

#include<iostream>

using namespace std;

int main()

{

int i,j,k;

cout<<endl;

for(i=0;i<4;i++)

{

for(j=0;j<4;j++) //Odd line

cout<<" \* ";

cout<<endl;

for(k=0;k<4;k++) //Even line

cout<<"\* ";

cout<<endl;

}

cout<<endl;

return 0;

}

/\*

Code for n\*n checkers

---------------------

#include<iostream>

using namespace std;

int main()

{

int n,i,j;

cin>>n;

cout<<endl;

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

{

if(i%2==0)

{

for(j=0;j<n/2;j++) //Odd line

cout<<" \* ";

cout<<endl;

}

else

{

for(j=0;j<(n+1)/2;j++) //Even line

cout<<"\* ";

cout<<endl;

}

}

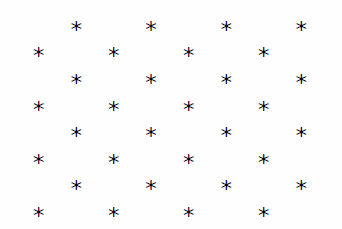
cou<<endl;

return 0;

}

\*/

Output



4 a)Write a C++ program that reads time in numeric form and displays it in English. The time is input as hours and minutes, separated by a space. Hours are specified in 24-hour time (15 is 3 P.M.), but the output should be in 12-hour A.M./P.M. form. Note that noon and midnight are special cases. Here are some examples:

Input1: Enter Time: 12 00 Output1: Noon

Input2: Enter time: 0 00 Output2: Midnight

Input3: Enter time: 6 44 Output3: Six forty four AM

Input4: Enter time: 18 11 Output4: Six eleven PM

#include<iostream>

#include<string>

using namespace std;

void time(int hours,int minutes)

{

int tens=0,ones;

if(minutes>20)

tens=(minutes/10)\*10;

ones=minutes-tens;

switch (hours)

{

case 0 : cout<<"Zero ";

break;

case 1 :

case 13: cout << "One ";

break;

case 2 :

case 14: cout << "Two ";

break;

case 3 :

case 15: cout << "Three ";

break;

case 4 :

case 16: cout << "Four ";

break;

case 5 :

case 17: cout << "Five ";

break;

case 6 :

case 18: cout << "Six ";

break;

case 7 :

case 19: cout << "Seven ";

break;

case 8 :

case 20: cout << "Eight ";

break;

case 9 :

case 21: cout << "Nine ";

break;

case 10:

case 22: cout << "Ten ";

break;

case 11:

case 23: cout << "Eleven ";

break;

case 12: cout << "Twelve ";

}

switch(tens)

{

case 20: cout << "twenty ";

break;

case 30: cout << "thirty ";

break;

case 40: cout << "forty ";

break;

case 50: cout << "fifty ";

}

switch(ones)

{

case 1: cout << "one ";

break;

case 2: cout << "two ";

break;

case 3: cout << "three ";

break;

case 4: cout << "four ";

break;

case 5: cout << "five ";

break;

case 6: cout << "six ";

break;

case 7: cout << "seven ";

break;

case 8: cout << "eight ";

break;

case 9: cout << "nine ";

break;

case 10: cout << "ten ";

break;

case 11: cout << "eleven ";

break;

case 12: cout << "twelve ";

break;

case 13: cout << "thirteen ";

break;

case 14: cout << "fourteen ";

break;

case 15: cout << "fifthteen ";

break;

case 16: cout << "sixteen ";

break;

case 17: cout << "seventeen ";

break;

case 18: cout << "eighteen ";

break;

case 19: cout << "nineteen ";

break;

}

if(hours<12)

cout<<"AM";

else

cout<<"PM";

}

int main()

{

int hours,minutes;

cout<<”Enter time:”;

cin>>hours>>minutes;

if(hours==12 && minutes==0)

cout<<"Noon";

else if(hours==0 && minutes==0)

cout<<"Midnight";

else if(hours>=0 && hours<=23 && minutes>=0 && minutes<=59)

time(hours,minutes);

else

cout<<"Invalid Format";

return 0;

}

Output:

Enter time:12 00

Noon

b) Beautiful String - Claudia is playing with strings. She has two strings and she wanted both of them to be beautiful. According to Claudia, both the strings are beautiful if one of the strings is a permutation of the other. Write a C++ program to determine whether the two given strings are beautiful or not.

Input Format: contains two string S and P.

Output Format: Print "YES" (without quotes) if the given strings are beautiful else print "NO" (without quotes).

Examples:

Input1: hello olelh Output1: YES

Input2: jgec cgec Output2: NO