Q. 01

```
// Write a program to convert
// binary or octal number
// into decimal number

#include <iostream>
using namespace std;
int main()
{
int num1, num2, remainder, originalNum, result = 0,quotient;
cout << "Press a number (1 for binary, 2 for octal): ";
cin >> num1;
switch (num1)
{
case 1:{
cout << "Enter a binary number (3,4 or 5 digits):"; // Take input
cin >> num2;

//If num is greater than 5-digit and less then 3-digit
if(originalNum <= 99 || originalNum > 11111) { cout << "Invalid input" <<endl; return 0;}
originalNum = num2;

//For 5th placed digit
if(originalNum > 1111){
quotient = num2 / 10000;
if (quotient != 0 && quotient != 1) { cout << "Invalid input"; return 0;}
result += (quotient * 16); // 5th placed digit from right (2^4)
quotient = (num2/1000) % 10; //Extract digit that is on 4rth placed from right
result += (quotient * 8); //Append in reuslt
}
//For 4rth placed digit
else if(originalNum > 111){
quotient = num2 / 1000;
if (quotient != 0 && quotient != 1) {cout << "Invalid input"; return 0;}
result += (quotient * 8); // 4th placed digit from right (2^3)
}
//For 1st placed digit
remainder = num2 % 10;
if (remainder != 0 && remainder != 1){cout << "Invalid input"; return 0; }
result += (remainder * 1); // 1st placed digit from right (2^0)
num2 = num2 / 10; //Removing first digit from right

//For 2nd placed digit
```

```
remainder = num2 % 10;
if (remainder != 0 && remainder != 1) {cout << "Invalid input"; return 0;}
result += (remainder * 2); // 2nd placed digit from right (2^1)
num2 = num2 / 10;

//For 3rd placed digit
remainder = num2 % 10;
if (remainder != 0 && remainder != 1) {cout << "Invalid input"; return 0;}
result += (remainder * 4); // 3rd placed digit from right (2^2)
cout << "Decimal notation of binary (" << originalNum << ") = " << result << endl;
break;
}
case 2:{ //For octal number

cout << "Enter Octal number (3,4 or 5 digits):"; // Take input
cin >> num2;
originalNum = num2;
if(originalNum <= 99 || originalNum > 77777){cout << "Invalid input" << endl; return 0;}

//For 5th placed digit
if(originalNum > 7777){
quotient = num2 / 10000;
if (quotient > 7) { cout << "Invalid input"; return 0;}
result += (quotient * 4096); // 5th placed digit from right (2^4)
quotient = (num2/1000) % 10; //Extract digit that is on 4rth placed from right
result += (quotient * 512); // 4rth placed digit from right (2^3)
}
//For 4rth placed digit
else if(originalNum > 777){
quotient = num2 / 1000;
if (quotient > 7) { cout << "Invalid input"; return 0;}
result += (quotient * 512); // 4th placed digit from right (2^3)
}
//For 1st placed digit
remainder = num2 % 10;
if (remainder > 7) { cout << "Invalid input"; return 0;}
result += (remainder * 1); // 1st placed digit from right (2^0)
num2 = num2 / 10;
//For 2nd placed digit
remainder = num2 % 10;
if (remainder > 7) { cout << "Invalid input"; return 0;}
result += (remainder * 8); // 2nd placed digit from right (2^1)
num2 = num2 / 10;
//For 3rd placed digit
remainder = num2 % 10;
if (remainder > 7) { cout << "Invalid input"; return 0;}
result += (remainder * 64); // 3rd placed digit from right (2^2)
cout << "Decimal notation of Octal (" << originalNum << ") = " << result << endl;
```

```
break;
}
default:
cout << "Number must be 1 or 2 any other input will considered invalid :)" << endl;
break;
}
}
```

Output Q1

```
Press a number (1 for binary, 2 for octal): 2
Enter Octal number (3,4 or 5 digits):76543
Decimal notation of Octal (76543) = 32099
[1] + Done                       "/usr/bin/gdb" --interpreter=mi
```

Q 02

```
// Monthly Income Tax Calculation for Pakistani Citizens

#include <iostream>
#include <iomanip>
using namespace std;
int main()
{
float taxOnAnnualIncome, annualIcome, originalAmount, rebate, monthlyTaxDeduct;
char salariedPerson, teacher;
bool Isteacher = true;
cout << fixed << setprecision(3);
cout << ">>>Tax calculation for fiscal year 2023-2024<<< \n" << endl;
cout << "Enter your annual income (PKR): ";
cin >> annualIcome;
cout << endl;
if(cin.fail() || annualIcome < 0){ //Input validation
cout << "Invalid annual income!" << endl;
return 1;
}
originalAmount = annualIcome;
cout << "Are you salaried person:" << endl;
cout << "Press 'Y' for Yes 'N' for NO: ";
cin >> salariedPerson;
cout << endl;

// check for invalid input
if (salariedPerson != 'y' && salariedPerson != 'Y')
```

```
if (salariedPerson != 'n' && salariedPerson != 'N'){
cout << "Invalid input" << endl;
return 1;
}
cout << "Are you a teacher in non-profit organization?" << endl;
cout << "Press 'Y' for Yes 'N' for No: ";
cin >> teacher;
cout << endl;

// check for invalid input
if (teacher != 'y' && teacher != 'Y')
if (teacher != 'n' && teacher != 'N'){
cout << "Invalid input" << endl;
return 1;
}
if (teacher != 'y' && teacher != 'Y')
Isteacher = false;

// For salaried and non-salaried persons
if (annualIcome <= 600000){
cout << "No Tax on this amount :)" << endl;
return 0;
}
// Tax Calculations for Salaried Individuals
if (salariedPerson == 'y' || salariedPerson == 'Y')
{
if (annualIcome <= 1200000){ // 600,001 - 1200,000
annualIcome = annualIcome - 600000;
taxOnAnnualIncome = annualIcome * (2.5 / 100);
cout << "For Salaried Individuals:" << endl;
if (!Isteacher)
cout << "Total Tax on annual income" << " = " << taxOnAnnualIncome << " PKR" << endl;
}
else if (annualIcome <= 2400000){ // 1,200,001 to PKR 2,400,000
annualIcome = annualIcome - 1200000;
taxOnAnnualIncome = 15000 + (annualIcome * (12.5 / 100));
cout << "For Salaried Individuals:" << endl;
if (!Isteacher)
cout << "Total Tax on annual income" << " = " << taxOnAnnualIncome << " PKR" << endl;
}
else if (annualIcome <= 3600000){ // 2,400,001 to PKR 3,600,000
annualIcome = annualIcome - 2400000;
taxOnAnnualIncome = 165000 + (annualIcome * (20.0 / 100));
cout << "For Salaried Individuals:" << endl;
if (!Isteacher)
cout << "Total Tax on annual income" << " = " << taxOnAnnualIncome << " PKR" << endl;
}
else if (annualIcome <= 6000000){ // 3,600,001 to PKR 6000,000
annualIcome = annualIcome - 3600000;
```

```
taxOnAnnualIncome = 405000 + (annualIcome * (25.0 / 100));
cout << "For Salaried Individuals:" << endl;
if (!Isteacher)
cout << "Total Tax on annual income" << " = " << taxOnAnnualIncome << " PKR" << endl;
}
else if (annualIcome > 6000000){ // Above 6,000,000
annualIcome = annualIcome - 6000000;
taxOnAnnualIncome = 1005000 + (annualIcome * (32.5 / 100));
cout << "For Salaried Individuals:" << endl;
if (!Isteacher)
cout << "Total Tax on annual income" << " = " << taxOnAnnualIncome << " PKR" << endl;
}
//Tax dedution for tecaher(salaried) working in non-profit organizaion
if (Isteacher){
cout << "Total Tax on annual income before rebate" << " = " << taxOnAnnualIncome << " PKR" << endl;
rebate = taxOnAnnualIncome * (25.0 / 100);
cout << "Tax rebated :" << rebate << endl;
taxOnAnnualIncome = taxOnAnnualIncome - rebate;
cout << "Total Tax on annual income (for teacher) after rebate" << " = " << taxOnAnnualIncome << " PKR" << endl;
}
// Monthly tax deduction
monthlyTaxDeduct = taxOnAnnualIncome / 12.0;
cout << "Monthly Tax deduction = " << monthlyTaxDeduct << endl;
}
// Tax Calculations for Non-Salaried Individuals
else if (salariedPerson == 'n' || salariedPerson == 'N')
{
if (annualIcome <= 1200000){ // 600,001 - 1200,000
annualIcome = annualIcome - 600000;
taxOnAnnualIncome = annualIcome * (5.0 / 100);
cout << "For Non-Salaried Individuals:" << endl;
if(!Isteacher)
cout << "Total Tax on annual income" << " = " << taxOnAnnualIncome << " PKR" << endl;
}
else if (annualIcome <= 2400000){ // 1,200,001 to PKR 2,400,000
annualIcome = annualIcome - 1200000;
taxOnAnnualIncome = 30000 + (annualIcome * (12.5 / 100));
cout << "For Non-Salaried Individuals:" << endl;
if(!Isteacher)
cout << "Total Tax on annual income" << " = " << taxOnAnnualIncome << " PKR" << endl;
}
else if (annualIcome <= 3600000){ // 2,400,001 to PKR 3,600,000
annualIcome = annualIcome - 2400000;
taxOnAnnualIncome = 180000 + (annualIcome * (17.5 / 100));
cout << "For Non-Salaried Individuals:" << endl;
if(!Isteacher)
cout << "Total Tax on annual income" << " = " << taxOnAnnualIncome << " PKR" << endl;
}
else if (annualIcome <= 6000000){ // 3,600,001 to PKR 6000,000
```

```
annualIcome = annualIcome - 3600000;
taxOnAnnualIncome = 390000 + (annualIcome * (22.5 / 100));
cout << "For Non-Salaried Individuals:" << endl;
if(!Isteacher)
cout << "Total Tax on annual income" << " = " << taxOnAnnualIncome << " PKR" << endl;
}
else if (annualIcome > 6000000){ // Above 6,000,000
annualIcome = annualIcome - 6000000;
taxOnAnnualIncome = 975000 + (annualIcome * (30.0 / 100));
cout << "For Non-Salaried Individuals:" << endl;
if(!Isteacher)
cout << "Total Tax on annual income" << " = " << taxOnAnnualIncome << " PKR" << endl;
}

//Tax deduction for tecaher (non-salaried) working in non-profit organizaion
if (Isteacher){
cout << "Total Tax on annual income before rebate" << " = " << taxOnAnnualIncome << " PKR" << endl;
rebate = taxOnAnnualIncome * (25.0 / 100);
cout << "Tax rebated :" << rebate << endl;
taxOnAnnualIncome = taxOnAnnualIncome - rebate;
cout << "Total Tax on annual income (for teacher)" << " = " << taxOnAnnualIncome << " PKR" << endl;
}

// Monthly tax deduction
monthlyTaxDeduct = taxOnAnnualIncome / 12.0;
cout << "Monthly Tax deduction = " << monthlyTaxDeduct << endl;
}
return 0;
}
```

Output Q2

```
>>>Tax calculation for fiscal year 2023-2024<<<

Enter your annual income (PKR): 4500000

Are you salaried person:
Press 'Y' for Yes 'N' for NO: y

Are you a teacher in non-profit organization?
Press 'Y' for Yes 'N' for No: n

For Salaried Individuals:
Total Tax on annual income = 630000.000 PKR
Monthly Tax deduction = 52500.000
```

Q 03

```cpp
// FESCO Residential Electricity Bill Calculation

#include <iostream>
#include<iomanip>
using namespace std;
int main()
{
int unitsConsumed;
float tax,totalBill;
cout << fixed << setprecision(3);
cout << "Enter number of units consumed:";
cin >> unitsConsumed;
if (cin.fail() || unitsConsumed < 0){
cout << "Invalid input total consumed units must be +ve integer value." << endl;
return 1;
}
// Bill calculation
// 0 - 100
if (unitsConsumed <= 100)
totalBill = unitsConsumed * 5.0;
else if (unitsConsumed <= 300)
totalBill = (100 * 5.0) + ((unitsConsumed - 100) * 7.0);
// 301 - 400
else if (unitsConsumed <= 400)
totalBill = (100 * 5.0) + (200 * 7.0) + ((unitsConsumed - 300) * 24.40);
// 401 - 500
else if (unitsConsumed <= 500)
totalBill = (100 * 5.0) + (200 * 7.0) + (100 * 24.40) + ((unitsConsumed - 400) * 24.91);
```

```
// 501 600
else if (unitsConsumed <= 600)
totalBill = (100 * 5.0) + (200 * 7.0) + (100 * 24.40) + (100 * 24.91) + ((unitsConsumed - 500) * 26.15);
// 601 - 700
else if (unitsConsumed <= 700)
totalBill = (100 * 5.0) + (200 * 7.0) + (100 * 24.40) + (100 * 24.91) + ((unitsConsumed - 600) * 27.59);
else{
totalBill = (100 * 5.0) + (200 * 7.0) + (100 * 24.40) + (100 * 24.91) + (100 * 27.56)+ ((unitsConsumed - 700) * 40);
}

// Total bill (before tax)
cout << "Total Bill (before tax) :" << totalBill << endl;

// Tax on total bill
tax = totalBill * (10.0 / 100);
cout << "Total tax :" << tax << endl;

// total bill (afer tax)
totalBill = totalBill + tax;
cout << "Total Bill (after tax):" << totalBill << endl;
return 0;
}
```

Output Q 3

```
Enter number of units consumed:596
Total Bill (before tax) :9341.400
Total tax :934.140
Total Bill (after tax):10275.540
```

Q 04

```
// Advanced Health Insurance Premium Calculator

#include <iostream>
using namespace std;
int main()
{
int age;
float premium,basePremium, smokeCharges, preConditions;
char smoking;
cout << ">>Advanced Health Insurance Premium Calculator<< \n" << endl;
cout << "Enter your age (Years): ";
```

```
cin >> age;
// Input validation
if (cin.fail() || age <= 0){
cout << "Invalid age!" << endl;
return 1;
}
cout << "Do you have any pre-existing conditions (diseases)? \n" << endl;
cout << "Press 0 if you don't,1 if you have one" << endl;
cout << "pre-existing condition and upto so on: ";
cin >> preConditions;
if (cin.fail() || preConditions < 0){ //Input validation
cout << "Invalid input!" << endl;
return 1;
}
cout << endl;
cout << "Do you somke?" << endl;
cout << "Press Y for 'Yes', N for 'No':";
cin >> smoking;
// Input validation
if (smoking != 'y' && smoking != 'Y')
if (smoking != 'n' && smoking != 'N'){
cout << "Invalid input" << endl;
return 1;
}
// Age of applicant
basePremium = (age < 18) ? 2000 : (age <= 30) ? 3500
                              : (age <= 50) ? 5000
                                 : 8000;
// Pre-existing conditions
if(preConditions == 1) premium = basePremium + basePremium * 0.25; //25% add in basePremium and assign to
premium
else if(preConditions > 1) premium = basePremium + basePremium * 0.5; //50% add in basePremium assign to
premium

// Smoker status
if(smoking == 'Y' || smoking == 'y') premium = premium + basePremium * 0.2; //20% of basePremium add in premium
cout << "Total premium is : PKR " << premium << " per month" << endl;
return 0;
}
```

Output Q4

```
>>Advanced Health Insurance Premium Calculator<<

Enter your age (Years): 43
Do you have any pre-existing conditions (diseases)?

Press 0 if you don't,1 if you have one
pre-existing condition and upto so on: 2

Do you somke?
Press Y for 'Yes', N for 'No':n
Total premium is : PKR 7500 per month
```

Q 05

```cpp
// Create a C++ program that assesses an individual's
// qualification for a personal loan

#include <iostream>
using namespace std;
int main()
{
float annualIncome, debitToIncomeRatio;
int creditScore, empDuration;

// Take annual income
cout << "Enter your annual income: ";
cin >> annualIncome;
if (cin.fail() || annualIncome < 0){ // input validation
cout << "Your annual income must be a +ve integer value" << endl;
return 1;
}
if (annualIncome < 1500000){ // for insufficient annual income
cout << endl;
cout << "You'r ineligible as your annual income is too low." << endl;
return 0;
}
// Take credit score
cout << "Enter your credit score: ";
cin >> creditScore;
if (cin.fail() || creditScore < 0){ // input validation
cout << "Your credit score must be +ve integer value" << endl;
return 1;
}
if (creditScore <= 700){ // for insufficient credit score
```

```
cout << endl;
cout << "You'r ineligible as your credit score is too low." << endl;
return 0;
}
// Take Debit-to-income ratio
cout << "Enter your debit-to-income ratio: ";
cin >> debitToIncomeRatio;
if (cin.fail() || debitToIncomeRatio < 0){ // input validation
cout << "Your Debit-to-income ratio must be a +ve integer value" << endl;
return 1;
}
if(debitToIncomeRatio > 35){
cout << endl;
cout << "You'r Debit-to-income ratio is too high. So you'r ineligible." << endl;
return 0;
}
// Employment status
cout << "Enter your employment duration (years)" << endl;
cout << "e.g for 2 years write 2, 3 years write 3: ";
cin >> empDuration;
if (cin.fail() || empDuration < 0){ // input validation
cout << "Invalid input!" << endl;
return 1;
}
if(empDuration < 2){
cout << endl;
cout << "Insufficient employment duration.So you'r ineligible for loan." << endl;
return 0;
}

// if person is eligible
cout << "\nCongratulations! you'r eligible for this loan." << endl;
return 0;
}
```

Output Q5

```
Enter your annual income: 1900000
Enter your credit score: 780
Enter your debit-to-income ratio: 34
Enter your employment duration (years)
e.g for 2 years write 2, 3 years write 3: 6

Congratulations! you'r eligible for this loan.
```

Q 06

```
// Secure Login System with Multi-Layer Authentication
#include<iostream>
using namespace std;
int main(){
int userName, password, OTP,userOTP, originalUserName,originalPass;
bool failedAttempts = true;
originalUserName = 105789;
originalPass = 100348;
cout << "Your user name is: " << originalUserName << endl;
cout << "Your password is: "<< originalPass << endl;
cout << endl;
cout << "Enter User Name : ";
cin >> userName;
if(cin.fail() || userName < 0){
cout << "Invalid user name. Must be a +ve integer " << endl;
return 1;
}
cout << "Password :";
cin >> password;
if(cin.fail() || password < 0){
cout << "Invalid password." << endl;
return 1;
}
if(userName != originalUserName || password != originalPass){
cout << "\n******Invalid user-name or password******"<< endl;
failedAttempts = false;
return 0;
}
else{
srand(time(0));
OTP = rand() % 10000; //Generating a 4-digit otp
cout << endl;
cout << "Your OTP is: " << OTP << endl;
cout << endl;
cout << "Plz enter your OTP: ";
cin >> userOTP;
if(userOTP == OTP){
cout<< "OTP verified!" << endl;
cout << "You have successfully loged into your account" << endl;
return 0;
}
else{
cout << "Wrong OTP!"<< endl;
failedAttempts = false;
return 0;
}
```

```
}
if(!failedAttempts){
cout << "Sorry, You can't login into your account for current session." << endl;
}
return 0;
}
```

Output Q6

```
Your user name is: 105789
Your password is: 100348

Enter User Name : 105789
Password :100348

Your OTP is: 791

Plz enter your OTP: 791
OTP verified!
You have successfully loged into your account
```

Q 07

```
// Conditional Decryption Scheme

#include <iostream>
using namespace std;
int main()
{
char char1, char2, char3, char4, char5, originalChar1, originalChar2;
cout << ">> Decryption Scheme << \n" << endl;
cout << "Enter five characters (only alphabates and integers) " << endl;
cout << "first character must be an alphabate (upper or lower case): ";
cin >> char1 >> char2 >> char3 >> char4 >> char5;

// Check first character must be an alphabate
if (!((char1 >= 'A' && char1 <= 'Z') || (char1 >= 'a' && char1 <= 'z'))){
cout << "First character must be an alphabate " << endl;
return 0;
}

// Input validaition for char2
if (!((char2 >= 'A' && char2 <= 'Z') || (char2 >= 'a' && char2 <= 'z') || (char2 >= '0' && char2 <= '9'))){
cout << "Invalid input! only alphabtes and integres are allowed." << endl;
```

```
return 0;
}
// Input validaition for char3
if (!((char3 >= 'A' && char3 <= 'Z') || (char3 >= 'a' && char3 <= 'z') || (char3 >= '0' && char3 <= '9'))){
cout << "Invalid input! only alphabtes and integres are allowed." << endl;
return 0;
}
// Input validaition for char4
if (!((char4 >= 'A' && char4 <= 'Z') || (char4 >= 'a' && char4 <= 'z') || (char4 >= '0' && char4 <= '9'))){
cout << "Invalid input! only alphabtes and integres are allowed." << endl;
return 0;
}
// Input validaition for char5
if (!((char5 >= 'A' && char5 <= 'Z') || (char5 >= 'a' && char5 <= 'z') || (char5 >= '0' && char5 <= '9'))){
cout << "Invalid input! only alphabtes and integres are allowed." << endl;
return 0;
}
// Scheme 1 (A - Z)
if(char1 >= 'A' && char1 <= 'Z'){

// Subtract 3 from each character's ASCII
char1 = char1 - 3;
char2 = char2 - 3;
char3 = char3 - 3;
char4 = char4 - 3;
char5 = char5 - 3;
originalChar1 = char1;
originalChar2 = char2;

// Reverse character's values
char1 = char5;
char2 = char4;
char4 = originalChar2;
char5 = originalChar1;
cout << "Decrypted message : " << char1 << char2 << char3 << char4 << char5 << endl;
}

// Scheme 2 (a - z)
else if(char1 >= 'a' && char1 <= 'z'){

//Vowel replacement:
// for char1
if(char1 == 'a' || char1 == 'A') char1 = char1 + 4;
else if(char1 == 'e' || char1 == 'E') char1 = char1 + 4;
else if(char1 == 'i' || char1 == 'I') char1 = char1 + 6;
else if(char1 == 'o' || char1 == 'O') char1 = char1 + 6;
else if(char1 == 'u' || char1 == 'U') char1 = char1 - 20;
```

```
// for char2
if(char2 == 'a' || char2 == 'A') char2 = char2 + 4;
else if(char2 == 'e' || char2 == 'E') char2 = char2 + 4;
else if(char2 == 'i' || char2 == 'I') char2 = char2 + 6;
else if(char2 == 'o' || char2 == 'O') char2 = char2 + 6;
else if(char2 == 'u' || char2 == 'U') char2 = char2 - 20;

// for char3
if(char3 == 'a' || char3 == 'A') char3 = char3 + 4;
else if(char3 == 'e' || char3 == 'E') char3 = char3 + 4;
else if(char3 == 'i' || char3 == 'I') char3 = char3 + 6;
else if(char3 == 'o' || char3 == 'O') char3 = char3 + 6;
else if(char3 == 'u' || char3 == 'U') char3 = char3 - 20;

// for char4
if(char4 == 'a' || char4 == 'A') char4 = char4 + 4;
else if(char4 == 'e' || char4 == 'E') char4 = char4 + 4;
else if(char4 == 'i' || char4 == 'I') char4 = char4 + 6;
else if(char4 == 'o' || char4 == 'O') char4 = char4 + 6;
else if(char4 == 'u' || char4 == 'U') char4 = char4 - 20;

// for char5
if(char5 == 'a' || char5 == 'A') char5 = char5 + 4;
else if(char5 == 'e' || char5 == 'E') char5 = char5 + 4;
else if(char5 == 'i' || char5 == 'I') char5 = char5 + 6;
else if(char5 == 'o' || char5 == 'O') char5 = char5 + 6;
else if(char5 == 'u' || char5 == 'U') char5 = char5 - 20;

// Swap Case:
if(char1 >= 'A' && char1 <= 'Z') char1 += 32;
else if(char1 >= 'a' && char1 <= 'z') char1 -= 32;

if(char2 >= 'A' && char2 <= 'Z') char2 += 32;
else if (char2 >= 'a' && char2 <= 'z') char2 -= 32;
if(char3 >= 'A' && char3 <= 'Z') char3 += 32;
else if (char3 >= 'a' && char3 <= 'z') char3 -= 32;
if(char4 >= 'A' && char4 <= 'Z') char4 += 32;
else if (char4 >= 'a' && char4 <= 'z') char4 -= 32;

if(char5 >= 'A' && char5 <= 'Z') char5 += 32;
else if (char5 >= 'a' && char5 <= 'z') char5 -= 32;

cout << "Decrypted message : " << char1 << char2 << char3 << char4 << char5 << endl;
}
return 0;
}
```

Output Q7

```
>> Decryption Scheme <<

Enter five characters (only alphabates and integers)
first character must be an alphabate (upper or lower case): Rt3Xe
Decrypted message : bU0qO
```

Q 08

```
// Smart Energy Billing System

#include <iostream>
using namespace std;

int main(){
int consumerNum, firstDigit;
float powerUsage, peakHourUsage,pHConsumption, finalBill;
cout << "Enter 5-digit consumer number: ";
cin >> consumerNum;

// Input validation
if(cin.fail() || consumerNum <= 9999 || consumerNum > 99999){
cout << "Invalid input! only 5-digit number is allowed.";
return 1;
}
cout << "Enter monthly power usage in kWh: ";
cin >> powerUsage;
// Input validation
if(cin.fail() || powerUsage < 0){
cout << "Invalid input! only 5-digit number is allowed.";
return 1;
}
firstDigit = consumerNum / 10000; // Left most digit

switch (firstDigit)
{

// Prime number (2, 3, 5, 7)
case 2:
case 3:
case 5:
case 7:
if(powerUsage < 300){
finalBill = powerUsage * 24.4;
```

```
finalBill -= finalBill * (10.0 / 100);
cout << "10% discount applied" << endl;
cout << "Final bill : " << finalBill << endl;
}
else if(powerUsage >= 300){
finalBill = powerUsage * 24.4;
cout << "No discounts and surcharges:" << endl;
cout << "Final bill : " << finalBill << endl;
}
break;

// Odd number but not prime (1, 9)
case 1:
case 9:
cout << "Enter your peak-hour consumption in kWh :";
cin >> peakHourUsage;
if(cin.fail() || peakHourUsage < 0){
cout << "Invalid input! peak hour always be a +ve value.";
return 1;
}
if(peakHourUsage > powerUsage){
cout << "Peak hour can't be greater than monthly power usage.";
return 0;
}
pHConsumption = (peakHourUsage / powerUsage) * 100;
if(pHConsumption > 50){
finalBill = powerUsage * 24.4;
finalBill += finalBill * (15.0 / 100);
cout << "15% surcharges applied." << endl;
cout << "Total bill : " << finalBill << endl;
}
else{
finalBill = powerUsage * 24.4;
cout << "No discounts or surcharges." << endl;
cout << "Final bill : " << finalBill << endl;
}
break;

// Even number (4, 6, 8)
case 0:
case 4:
case 6:
case 8:
finalBill = powerUsage * 24.4;
cout << "No discounts or surcharges." << endl;
cout << "Final bill : "<< finalBill << endl;
break;
}
return 0;
```

```
}
```

Output Q8

```
Enter 5-digit consumer number: 76543
Enter monthly power usage in kWh: 600
No discounts and surcharges:
Final bill : 14640
```

Q 09

```
// Recursive Rightmost Digit Manipulation

#include <iostream>
using namespace std;

int main()
{
int num, rightMostDigit, remainingNum;
cout << "Enter a 6-digit number: ";
cin >> num;
if(cin.fail() || num <= 99999 || num > 999999 ){
cout << "Invalid input. Number must be 6-digit integer." << endl;
return 0;
}

// For 6-digit (right most)
rightMostDigit = num % 10; //Extracting the right most digit
remainingNum = num / 10; // Removing the right most digit

if(rightMostDigit % 2 == 0){ // For even
remainingNum += 1;
if(remainingNum > 99999){ // If overflow one digit
remainingNum -= 2;
}
}
else if(rightMostDigit % 2 != 0){ // For odd
remainingNum -= 1;
}

// For 5th-digit (form left)
rightMostDigit = remainingNum % 10;
remainingNum /= 10;
if(rightMostDigit % 2 == 0){
remainingNum += 1;
if(remainingNum > 9999){
remainingNum -= 2;
}
```

```
}
else if(rightMostDigit % 2 != 0){ // For odd
remainingNum -= 1;
}

// For 4th -digit(from left)
rightMostDigit = remainingNum % 10;
remainingNum /= 10;

if(rightMostDigit % 2 == 0){
remainingNum += 1;
if(remainingNum > 999){
remainingNum -= 2;
}
}
else if(rightMostDigit % 2 != 0){ // For odd
remainingNum -= 1;
}

// For 3rd-digit(from left)
rightMostDigit = remainingNum % 10;
remainingNum /= 10;

if(rightMostDigit % 2 == 0){
remainingNum += 1;
if(remainingNum > 99){
remainingNum -= 2;
}
}
else if(rightMostDigit % 2 != 0){ // For odd
remainingNum -= 1;
}

// For 2nd-digit(from left)
rightMostDigit = remainingNum % 10;
remainingNum /= 10;

if(rightMostDigit % 2 == 0){
remainingNum += 1;
if(remainingNum > 9){
remainingNum -= 2;
}
}
else if(rightMostDigit % 2 != 0){ // For odd
remainingNum -= 1;
}
cout << "Final Result is: " << remainingNum << endl;
return 0;
```

```
}
```

Output Q9

```
Enter a 6-digit number: 123456
Final Result is: 0
```

Q 10

```
// Trip Planner for Weather Conditions

#include<iostream>
#include <iomanip>
using namespace std;

int main(){
char weather;
float temperature, wind;
cout << "Enter Weather forcast" << endl;
cout << "(s : sunny, r : rainy, n : snowy, f : foggy ) : ";
cin >> weather;
if (cin.fail() || (weather != 's' && weather != 'r' && weather != 'n' && weather != 'f')){
cout << "Invalid input" << endl;
return 0;
}
cout << "Enter temperature in degree celsius: ";
cin >> temperature;
if(cin.fail()){
cout << "Invalid input";
return 0;
}
cout << "Enter wind speed in km/h: ";
cin >> wind;
if(cin.fail()){
cout << "Invalid input";
return 0;
}
if(weather != 's' || temperature < 10 || temperature > 40 || wind > 50){
cout << "\nWeather conditions are not appropriate for out door trip. " << endl;
cout << endl;
cout << setw(49) << "<*><*><*> Stay home, Stay safe! <*><*><*>\n" << endl;
}
else
cout << "\nHurrah! you can go for out door trip as weather conditions are valid :)" << endl;
return 0;

}
```

Output Q10

```
Enter Weather forcast
(s : sunny, r : rainy, n : snowy, f : foggy ) : s
Enter temperature in degree celsius: 45
Enter wind speed in km/h: 55

Weather conditions are not appropriate for out door trip.

       <*><*><*> Stay home, Stay safe! <*><*><*>
```

The End 😊