**Homework** **(**C++ Fundamentals 1 (Types, Statements, Loops)**)**

1. ***Write a C++ program that asks the user to enter a positive integer, and then prints the factorial of that number.***

***Example  
Input:  
    5  
Output:  
    Factorial of 5 is 120***

#include <iostream>

//1st task: Factorial.

int main() {

unsigned int number;

std::cout <<"Enter a positive integer: ";

std::cin >> number;

unsigned int factorial = 1;

for(int i=number; i>0; --i){

factorial \*= i;

}

std::cout << "Factorial of " << number << " is " << factorial;

return 0;

}

1. ***Write a C++ program that takes an integer input from the user and calculates the sum of its digits.***

#include <iostream>

#include <cstdlib>

using namespace std;

//2nd task: Sum of digits.

int main() {

int number;

cout <<"Enter a number: ";

cin >> number;

number = abs(number);

int sum = 0;

int digit;

while (number != 0){

digit = number % 10;

number /= 10;

sum += digit;

}

cout << "Sum of digits: " << sum;

return 0;

}

1. ***Write a simple calculator program in C++ that takes two numbers and an operator (+, -, \*, /) as input.***

#include <iostream>

using namespace std;

//3rd task: Calculator.

int main() {

int number1, number2;

char op;

cout <<"Enter two numbers and operator: ";

cin >> number1 >> number2 >> op;

int result = 0;

switch (op) {

case '+': result = number1 + number2;

break;

case '-': result = number1 - number2;

break;

case '\*': result = number1 \* number2;

break;

case '/':

if(number2 == 0)

cout << "Division by 0 is not allowed.";

else result = number1 / number2;

break;

}

if (result == 0)

cout << "The entered operator was not found." << result;

else cout << "Result: " << result;

return 0;

}

1. ***Write a C++ program that takes an integer as input and determines whether it is a prime or composite number.***

#include <iostream>

#include <cmath>

using namespace std;

//4th task: Prime or Composite number.

int main() {

int number;

cout <<"Enter a number: ";

cin >> number;

if(number <= 1)

cout << number << " is not a prime or composite number.";

else if(number == 2)

cout << number << " is a prime number.";

else {

bool isComposite = false;

int limit = sqrt(number);

for(int i = 2; i <= limit; i++){

if(number % i == 0){

isComposite = true;

break;

}

}

if(isComposite == true)

cout << number << " is a composite number.";

else

cout << number << " is a prime number.";

}

return 0;

}

1. ***Write a program that will input an integer number and calculate the inverse of the number. Print the output.***

#include <iostream>

using namespace std;

//5th task: Inverse number.

int main() {

int number;

cout <<"Enter a number: ";

cin >> number;

int invNumber = 0;

while (number != 0){

invNumber = (invNumber \* 10) + (number % 10);

number /= 10;

}

// 2nd variant

/\* for(; number != 0; ){

invNumber = (invNumber \* 10) + (number % 10);

number /= 10;

}\*/

cout << "Inverse number: " << invNumber;

return 0;

}

1. ***Write a program that will input two integers - N and M. The program should "draw" a rectangle of NxM size using "\*", "/", "\" and spaces.***

#include <iostream>

using namespace std;

//6th task: "draw" a rectangle of NxM size using "\*", "/", "\" and spaces:

/\*\*\*\*\

\* \*

\* \*

\\*\*\*\*/

int main() {

int N, M;

cout << "Enter two integers, width (N) and height (M): ";

cin >> N >> M;

if (N < 2 || M < 2) {

cout << "Width and height must be at least 2 to form a rectangle.";

return 0;

}

for(int row = 0; row < M;++ row) {

for(int col = 0; col < N; col++) {

if(row == 0) {

if (col == 0) cout << '/';

else if (col == N - 1) cout << '\\'; // to understand that '/' sign

else cout << '\*';

}

else if(row == M - 1) {

if (col == 0) cout << '\\'; // to understand that '/' sign

else if (col == N - 1) cout << '/';

else cout << '\*';

}

else { // Middle rows

if (col == 0 || col == N - 1) cout << '\*';

else cout << ' ';

}

}

cout << endl;

}

return 0;

}

1. ***Write a program that calculates a random number from 1 through 100. The program then asks the user to guess the number. If the user guesses too high or too low, the program should output "too high" or "too low" accordingly.  
     
   The program must let the user continue to guess until the user correctly guesses the number. Output how many guesses it took the user to guess the correct number.  
     
     
   Note: Use the standard library function to generate the random number: http://en.cppreference.com/w/cpp/numeric/random/rand***

#include <iostream>

#include <cstdlib>

using namespace std;

//7th task: Guess the random number.

// rand() => get the same number every time you run the program: pseudo-random number generator.

// srand() => diferente results

int main() {

srand(time(0));

// time(0) returns the current time in seconds (as an integer).

int num = (rand() % 100) + 1;

// first part: 0 to 99, then plus 1, and will be 1 to 100

int guessNum = 0;

int cnt = 0;

while(num != guessNum) {

cnt++;

cout << "Guess the number: ";

cin >> guessNum;

if(guessNum > num)

cout << "too high" << endl;

else if(guessNum < num)

cout << "too low" << endl;

}

cout << "Wow, you guessed the right number!" << endl;

cout << "Guesses: " << cnt << endl;

return 0;

}

1. ***Write a program to input a number and convert it to binary  
   Example:  
       Input:   
           5  
       Output:   
           101***

#include <iostream>

#include <string> // for std::string

#include <algorithm> // for std::reverse

using namespace std;

//8th task: Convert to binary.

int main() {

unsigned int num;

cout << "Enter a number: ";

cin >> num;

if (num == 0) {

cout << "Binary: 0";

return 0;

}

string bin = "";

while (num > 0) {

bin += to\_string(num % 2);

num /= 2;

}

reverse(bin.begin(), bin.end());

cout << "Binary: " << bin << endl;

return 0;

}

1. ***Assume the following variable declarations on a 64-bit system***

***char a;  
int b;  
float c;  
double d;  
long e;  
short f;***

***How many bytes of memory are required for each variable, and what is the total memory used by all variables?***

char a; => 1 byte

int b; => 4 byte

float c; => 4 byte

double d; => 8 byte

long e; => 8 byte

short f; => 2 byte

Ընդհանուր՝ 1+4+4+8+8+2 = 27 bytes

1. ***In C++, consider the following code:***

***float f = 3.14f;  
int i = static\_cast<int>(f);***

***a) What happens in memory when static\_cast is used in this case?  
b) What value will be stored in i, and why?***

a) CPU-ն լողացող կետով թիվը փոխակերպում է ամբողջ թվի՝ տասնորդական մասը կրճատելով (ոչ թե կլորացնելով)։

Սա նշանակում է, որ float թվի երկուական ներկայացումը մեկնաբանվում է որպես թիվ, իսկ կոտորակային մասը՝ անտեսվում։

Արդյունքում ստացված ամբողջ թվի բիթերը պահվում են i-ում որպես ստանդարտ 32-բիթանոց ամբողջ թիվ։

b) Կդառանա, 3.14f → 3

// i == 3

1. ***What happens if you divide an integer by zero in C++? What about dividing a floating-point number by zero?***

1) Այն առաջացնում է անորոշ վարքագիծ:

Համակարգերի մեծ մասում ծրագիրը.

Crash with a runtime error (e.g., divide-by-zero exception or SIGFPE)

Or behave unpredictably (compiler might optimize it away!)

Ամբողջ թվերի բաժանելուց առաջ միշտ պետք է ստուգեք զրոյի առկայությունը:

2) This does not crash.

C++ օգտագործում է IEEE 754 կանոնները for floating-point math.

Օրինակ՝ 5.0 / 0.0 => +inf

-5.0 / 0.0 => -inf

0.0 / 0.0 => NaN (Not a Number)

1. ***What is the result of the following expression and why?***

***int x = 5;  
int y = x++ + ++x;***

Պատասխանը կլինի 12։

Որովհետև, x++ անելուց գումարման մեջ օգտագործվելու է 5 արժեքը, բայց x դառնում է 6։ ++x անելուց 6-ը դառնում է 7, ւ հետո կատարվում գումարում՝ 5+7 = 12: