Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Лабораторная работа № 2

«Операторы цикла.»

Выполнил: Проверила:

студ. гр. 250504(пг.2) Скиба И. Г.

Пигулевский К. С.

МИНСК 2022

**Задача 1**

3. В 1626г. индейцы продали остров за 20 долларов. Если бы эти деньги были помещены в банк под 4% годовых (процент капитализированный), то какова была бы стоимость капитала сегодня?

Блок-схема алгоритма:



Исходный код:

int main()

{

float costOfCapitalToday = 20.f;

for (int i = 0; i < (2022 - 1626); i++)

costOfCapitalToday += costOfCapitalToday \* 0.04f;

printf("If in 1626 $20 were deposited in a bank at 4%% per annum, today the cost of capital would be - $%.0f\n", costOfCapitalToday);

}

Результат выполнения программы:



**Задача 2**

3. Составить алгоритм, определяющий, сколько существует способов набора одного рубля при помощи монет достоинством 50коп., 20коп., 5коп. и 2коп.

Блок-схема алгоритма:



Исходный код:

int main()

{

int i = 0;

int countOfKopecksInRubel = 100;

for (int a = 0; a <= 2; a++)

for (int b = 0; b <= 5; b++)

for (int c = 0; c <= 20; c++)

for (int d = 0; d <= 50; d++)

if (countOfKopecksInRubel == (a \* 50 + b \* 20 + c \* 5 + d \* 2))

i++;

printf("\nThe number of ways to collect one ruble from kopecks with a face value of 50 kop., 20kop., 5kop., 2kop. - %i\n", i);

}

Результат выполнения программы:



**Задача 3**

3. Даны положительные числа *A* и *B* (*A* > *B*). На отрезке длины *A* размещено максимально возможное количество отрезков длины *B* (без наложений). Не используя операции умножения и деления, найти количество отрезков *B*, размещенных на отрезке *A.*

Блок-схема алгоритма:



Исходный код:

int main()

{

float a = 0, b = 0;

printf("\nEnter length of 'A' and 'B' segments (A > B): ");

while (scanf\_s("%f %f", &a, &b) < 2 || (a <= 0 || a <= b))

throwErrorAndClearInputBuffer();

float tempSumOfBinA = 0;

int i = 0;

while (tempSumOfBinA < a)

{

tempSumOfBinA += b;

if (tempSumOfBinA <= a)

i++;

}

printf("Segment 'B' fits in segment 'A' %i times\n", i);

}

Результат работы программы:

