

Лабораторна робота №5. Циклічні конструкції

1 Вимоги

1.1 Розробник

- Бельчинська Катерина Юріївна
- студентка групи КІТ-320
- 04-nov-2020

1.2 Індивідуальне завдання

Визначити, чи є задане 6-значне число “щасливим” квитком (“щасливий квиток” – квиток, в якому сума першої половини чисел номера дорівнює сумі другої половини. Наприклад, білет з номером 102300 є щасливим, бо $1+0+2=3+0+0$). Реалізувати програму за допомогою трьох циклів: for, while-do, do-while.

2 Опис роботи

Код програми:

```
define NUM 123456
int main () {
    int result_for = 0;
    int num123_for = 0;
    int num456_for = 0;
    int result_while = 0;
    int num123_while = 0;
    int num456_while = 0;
    int result_do = 0;
    int num123_do = 0;
    int num456_do = 0;
    int half = NUM % 1000;
    for (int i = 1000; (NUM / i) > 0; i *= 10) { /* цикл відокремлює перші три цифри даного числа
та сумує їх, знаходимо перше число "щасливого" квитку */
        num123_for += NUM / i % 10;
    }
    for (int j = 1; (half / j) > 0; j *= 10) { /* цикл відокремлює останні три цифри даного числа
та сумує їх, знаходимо друге число "щасливого" квитку*/
        num456_for += half / j % 10;
    }
    if (num123_for == num456_for) { // порівняння двох отриманих чисел
        result_for = 1; // якщо квиток "щасливий"
    } else {
        result_for = -1; // якщо квиток не "щасливий"
    }

    int temp1_while = 1000; // змінна, що відокремлює перші три цифри даного числа, щоб сумувати
    їх у циклі
    while ((NUM / temp1_while) > 0) {
        num123_while += NUM / temp1_while % 10;
        temp1_while *= 10;
    }
    int temp2_while = 1; // змінна, що відокремлює останні три цифри даного числа, щоб сумувати їх
    у циклі
    while ((half / temp2_while) > 0) {
        num456_while += half / temp2_while % 10;
        temp2_while *= 10;
    }
    if (num123_while == num456_while) { // порівняння двох отриманих чисел
        result_while = 1; // якщо квиток "щасливий"
    } else {
```

```

        result_while = -1; // якщо квиток не "щасливий"
    }

    int temp1_do = 1000; // змінна, що відокремлює перші три цифри даного числа, щоб сумувати їх у
циклі
    do {
        num123_do += NUM / temp1_do % 10;
        temp1_do *= 10;
    } while ((NUM / temp1_do) > 0);
    int temp2_do = 1; // змінна, що відокремлює останні три цифри даного числа, щоб сумувати їх у
циклі
    do {
        num456_do += half / temp2_do % 10;
        temp2_do *= 10;
    } while ((half / temp2_do) > 0);
    if (num123_do == num456_do) { // порівняння двох отриманих чисел
        result_do = 1; // якщо квиток "щасливий"
    } else {
        result_do = -1; // якщо квиток не "щасливий"
    }
    return 0;
}

```

Компіляція програми :

```

kate@kate-K54C:~/Programming-Belchinskaya/lab05/3$ make clean prep compile
rm -rf dist
mkdir dist
gcc -std=gnu11 -g -Wall -Wextra -Wformat-security -Wfloat-equal -Wshadow -Wconversion -Wlogical-not-parentheses -Wnull-dereference -I./src src
c/main.c -o ./dist/main.bin
src/main.c: In function 'main':
src/main.c:9:6: warning: variable 'result_do' set but not used [-Wunused-but-set-variable]
  9 |     int result_do = 0;
    |     ^
src/main.c:6:6: warning: variable 'result_while' set but not used [-Wunused-but-set-variable]
  6 |     int result_while = 0;
    |     ^
src/main.c:3:6: warning: variable 'result_for' set but not used [-Wunused-but-set-variable]
  3 |     int result_for = 0;
    |     ^

```

Відладка програми:

```

23         result_for = -1; // якщо квиток не "щасливий"
24     }
25
26     int temp1_while = 1000; // змінна, що відокремлює перші три цифри даного чи
27     while ((NUM / temp1_while) > 0) {
28         num123_while += NUM / temp1_while % 10;
29         temp1_while *= 10;
30     }
31     int temp2_while = 1; // змінна, що відокремлює останні три цифри даного чис
32     while ((half / temp2_while) > 0) {
33         num456_while += half / temp2_while % 10;
34         temp2_while *= 10;
35     }
36     if (num123_while == num456_while) { // порівняння двох отриманих чисел
37         result_while = 1; // якщо квиток "щасливий"
38     } else {
39         result_while = -1; // якщо квиток не "щасливий"

```

Строка: 32, Столбец: 1

№	Переменная	Значение	Тип
10	num123_for	6	int
	num456_for	15	int
	result_while	0	int
	num123_while	6	int
	num456_while	15	int

3 Варіанти використання.

Програму доцільно використовувати для визначення “щасливості” квитка.

Висновок

Отже, визначено “щасливий” квиток за допомогою трьох типів циклів: for, while do, do while.

4 Загальне завдання

1

```
#define X 16// задане число
#define Y 20// задане число
int main () {
    int num1;
    int num2;
    int nod_for = 0;
    int nod_while = 0;
    int nod_do = 0;
    for (num1 = X, num2 = Y; num1 > 0 && num2 > 0;) { /*ділимо більше число на менше доки остача
від ділення не буде 0 (алгоритм Евкліда)*/

        (num1 > num2) ? (num1 %= num2) : (num2 %= num1);
        nod_for = num1 + num2;
    }
    num1 = X; /*щоб увійти у новий цикл, минулі значення змінних необхідно "скинути"
    num2 = Y;
    while (num1 > 0 && num2 > 0) { /*ділимо більше число на менше доки остача від ділення не буде
0*/
        if (num1 > num2) {
            num1 %= num2;
        } else if (num1 < num2) {
            num2 %= num1;
        }
        nod_while = num1 + num2;
    }
    num1 = X;
    num2 = Y;
    do { /*ділимо більше число на менше доки остача від ділення не буде 0*/

        if (num1 > num2) {
            num1 %= num2;
        } else if (num1 < num2) {
            num2 %= num1;
        }
        nod_do = num1 + num2;
    } while (num1 > 0 && num2 > 0);
    return 0;
}
```

2

```
#define NUM 3//задане число
int main () {
    int tempFor = 0;
    int resultFor = 0;
    for (int i = 2; i < NUM; i++) { /*якщо число ділиться націло (тобто остача від ділення =0) на
число менше за нього, то воно складене*/
        tempFor = NUM % i;
        if (tempFor == 0) {
            resultFor = 0; /*число складене
        } else {
            resultFor = 1; /*число просте
        }
    }
    int i = 2;
    int resultWhile = 0;
```

```

int tempWhile = 0;
while (i < NUM) {
    tempWhile = NUM % i;
    if (tempWhile == 0) {
        resultWhile = 0; //число складене
    } else {
        resultWhile = 1; //число просте
    }
    i++;
}
int j = 2;
int resultDo = 0;
int tempDo = 0;
do {
    tempDo = NUM % j;
    if (tempDo == 0) {
        resultDo = 0; //число складене
    } else {
        resultDo = 1; //число просте
    }
    j++;
} while (j < NUM);
return 0;
}

```

4

```

#define NUM 6 //задане число
int main () {
    int sum = 0;
    int temp_for = 0;
    int result_for = 0;
    for (int i = 1; i < NUM; i++) { //ділимо дане число націло (тобто остача = 0)
        temp_for = NUM % i;
        if (temp_for == 0) { //якщо остача від ділення 0 - сумуємо дільники
            sum += i;
        }
    }
    if (sum == NUM) {
        result_for = 1; //число досконале
    }
    else {
        result_for = 0; //число не досконале
    }
    int sum_while = 0;
    int temp_while = 0;
    int result_while = 0;
    int step_while = 1; //ділимо дане число націло
    while (step_while < NUM) {
        temp_while = NUM % step_while;
        if (temp_while == 0) { //якщо остача від ділення 0 - сумуємо дільники
            sum_while += step_while;
        }
        step_while++;
    }
    if (sum == NUM) {
        result_while = 1; //число досконале
    }
    else {
        result_while = 0; //число не досконале
    }

    int sum_do = 0;
    int temp_do = 0;
    int result_do = 0;
    int step_do = 1;
    do {
        temp_do = NUM % step_do;
        if (temp_do == 0) {
            sum_do += step_do;
        }
    }

```

```

        step_do++;
    } while (step_do < NUM);
    if (sum == NUM) {
        result_do = 1; //число досконале
    }
    else {
        result_do = 0; //число не досконале
    }

    return 0;
}

```

5

```

#define X 25
int main () {
    int num1 = X;
    int rootFor = 1;
    int rootWhile = 1;
    int rootDo = 1;
    int tempWhile = 0;
    int tempDo = 0;
    for (int tempFor = 0; tempFor != rootFor;) {
        tempFor = num1 / rootFor;
        if (tempFor == rootFor) {
            break;
        }
        rootFor++;
    }
    int resultFor = -1;
    if (rootFor * rootFor == num1) {
        resultFor = rootFor;
    }
    num1 = X;
    while (tempWhile != rootWhile) {
        tempWhile = num1 / rootWhile;
        if (tempWhile == rootWhile) {
            break;
        }
        rootWhile++;
    }
    int resultWhile = -1;
    if (rootWhile * rootWhile == num1) {
        resultWhile = rootWhile;
    }
    num1 = X;
    do {
        tempDo = num1 / rootDo;
        if (tempDo == rootDo) {
            break;
        }
        rootDo++;
    } while (tempDo != rootDo);
    int resultDo = -1;
    if (rootDo * rootDo == num1) {
        resultDo = rootDo;
    }
    return 0;
}

```

5

6