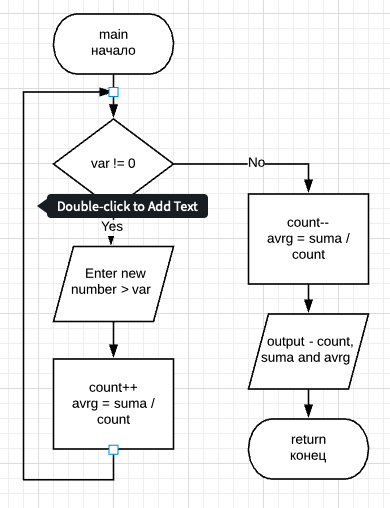
# Организовать беспрерывный ввод чисел с клавиатуры, пока пользователь не введёт 0. После ввода нуля, показать на экран количество чисел, которые были введены, их общую сумму и среднее арифметическое.



#include <iostream>

int main() {

double var;

double count = 0;

double suma = 0;

double avrg;

while(var != 0) {

std::cout << "\rEnter new number: ";

std::cin >> var;

count++;

suma += var;

}

count--;

avrg = suma / count;

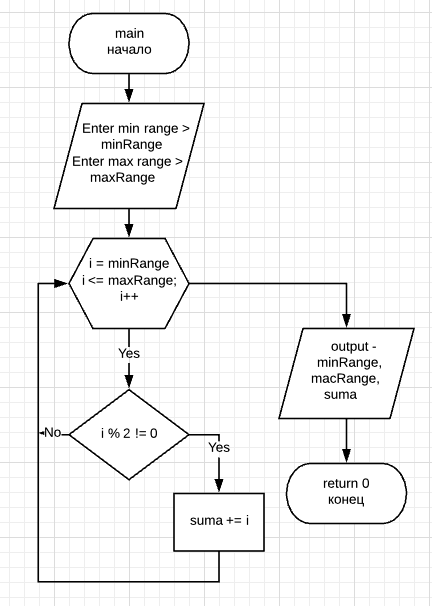
std::cout << "\nCount - " << count << "\nSuma - " << suma;

std::cout << "\nAverage - " << avrg;

return 0;

}

# Необходимо суммировать все нечётные целые числа в диапазоне, который введёт пользователь с клавиатуры.



#include <iostream>

int main() {

int suma = 0;

int minRange;

int maxRange;

std::cout << "Enter min range: ";

std::cin >> minRange;

std::cout << "Enter max range: ";

std::cin >> maxRange;

for(int i = minRange; i <= maxRange; i++)

if(i % 2 != 0)

suma += i;

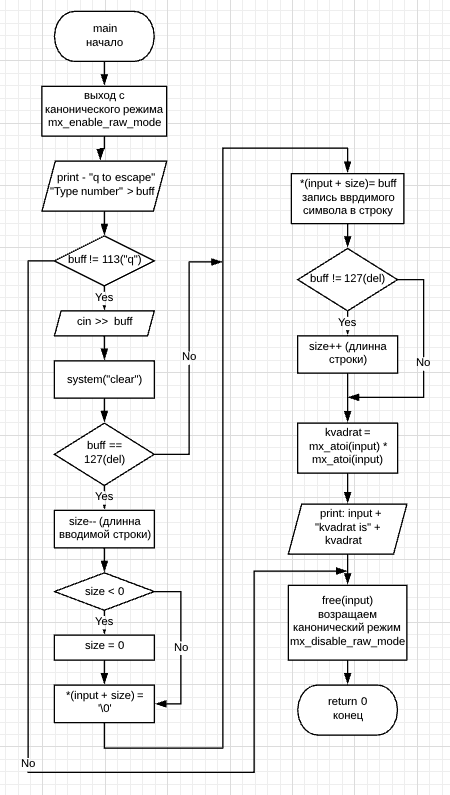
std::cout << "Sum of even numbers in range ["

<< minRange << ", " << maxRange << "] is = " << suma;

return 0;

}

Написать программу, которая будет показывать на экран квадрат числа, введенного пользователем. Пользователь должен сам решать — выйти из программы или продолжить ввод



#include <iostream>

#include <termios.h>

#include <stdlib.h>

#include <sys/ioctl.h>

static int mx\_atoi\_add(const char\* str) {

int result = 0;

for (int i = 0; \*(str + i) != '\0'; ++i)

result = result \* 10 + \*(str + i) - '0';

return result;

}

static int mx\_atoi(const char \* str) {

int str\_len = strlen(str);

char \*num\_str = (char \*)malloc(str\_len \* sizeof(char));

int index = 0;

char ch;

if (!str)

return 0;

for (int i = 0; i < str\_len; ++i) {

ch = \*(str + i);

if (ch == 0) {

\*(num\_str + index) = 0;

break;

}

if (isdigit(ch))

\*(num\_str + index++) = ch;

}

\*(num\_str + index) = 0;

int ret = mx\_atoi\_add((const char \*)num\_str);

free(num\_str);

return ret;

}

/\*

\* Changes terminal`s mode to raw.

\*/

static void mx\_enable\_raw\_mode(struct termios orig\_termios) {

struct termios raw = orig\_termios;

tcgetattr(0, &orig\_termios);

raw.c\_lflag &= ~(ICANON | ECHO);

tcsetattr(0, TCSAFLUSH, &raw);

}

/\*

\* Changes terminal`s mode to original.

\*/

static void mx\_disable\_raw\_mode(struct termios orig\_termios) {

tcsetattr(0, TCSAFLUSH, &orig\_termios);

}

static char \*mx\_strnew(int size) {

int sized = size \* 2;

char \*memory = (char\*)malloc(sized + 1);

if (memory == NULL)

return NULL;

for (int i = 0; i < sized + 1; i++)

\*(memory + i) = '\0';

return memory;

}

int main() {

struct termios orig\_termios;

char buff = '\0';

int size = 0;

char\* input = mx\_strnew(4096);

int kvadrat;

tcgetattr(0, &orig\_termios);

mx\_enable\_raw\_mode(orig\_termios);

std::cout << "q - to escape\nType number";

while(buff != 113) {

std::cin >> buff;

system("clear");

if(buff == 127) {

size--;

if(size < 0)

size = 0;

\*(input + size) = '\0';

}

\*(input + size)= buff;

if(buff != 127)

size++;

kvadrat = mx\_atoi(input) \* mx\_atoi(input);

std::cout << "\r" << input << " kvadrat is - "

<< kvadrat;

}

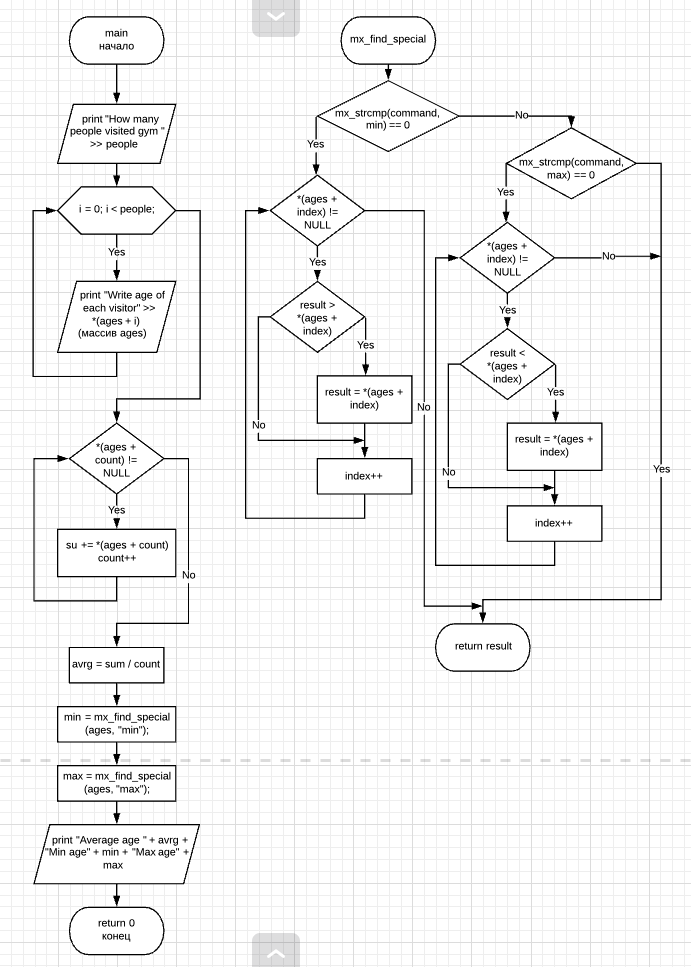
free(input);

mx\_disable\_raw\_mode(orig\_termios);

return 0;

}

# В спортзал ежедневно приходит какое-то количество посетителей. Необходимо предложить пользователю ввести такие данные: сколько человек посетило спортзал за день, ввести возраст каждого посетителя и в итоге показать возраст самого старшего и самого молодого из них, а так же посчитать средний возраст посетителей.



#include <iostream>

static int mx\_strcmp(const char \*s1, const char \*s2) {

int counter = 0;

while (\*(s1 + counter) == \*(s2 + counter)) {

if (\*(s1 + counter) == '\0' && \*(s2 + counter) == '\0')

return 0;

counter++;

}

return s1[counter] - s2[counter];

}

static int mx\_find\_special(int\* ages, char\* command) {

int result = \*ages;

int index = 0;

if(!mx\_strcmp(command, "min"))

while(\*(ages + index)) {

if(result > \*(ages + index))

result = \*(ages + index);

index++;

}

else if(!mx\_strcmp(command, "max"))

while(\*(ages + index)) {

if(result < \*(ages + index))

result = \*(ages + index);

index++;

}

return result;

}

static int mx\_get\_avrg(int\* ages) {

int sum = 0;

int count = 0;

int avrg;

while(\*(ages + count)) {

sum += \*(ages + count);

count++;

}

avrg = sum / count;

return avrg;

}

int main() {

int people;

int\* ages;

int avrg;

int min;

int max;

std::cout << "How many people visited gym? \nAnswer: ";

std::cin >> people;

ages = (int\*)malloc(sizeof(int) \* people);

for(int i = 0; i < people; i++) {

std::cout << "Write age of each visitor: ";

std::cin >> \*(ages + i);

}

avrg = mx\_get\_avrg(ages);

min = mx\_find\_special(ages, (char\*)"min");

max = mx\_find\_special(ages, (char\*)"max");

std::cout << "Average age is - " << avrg;

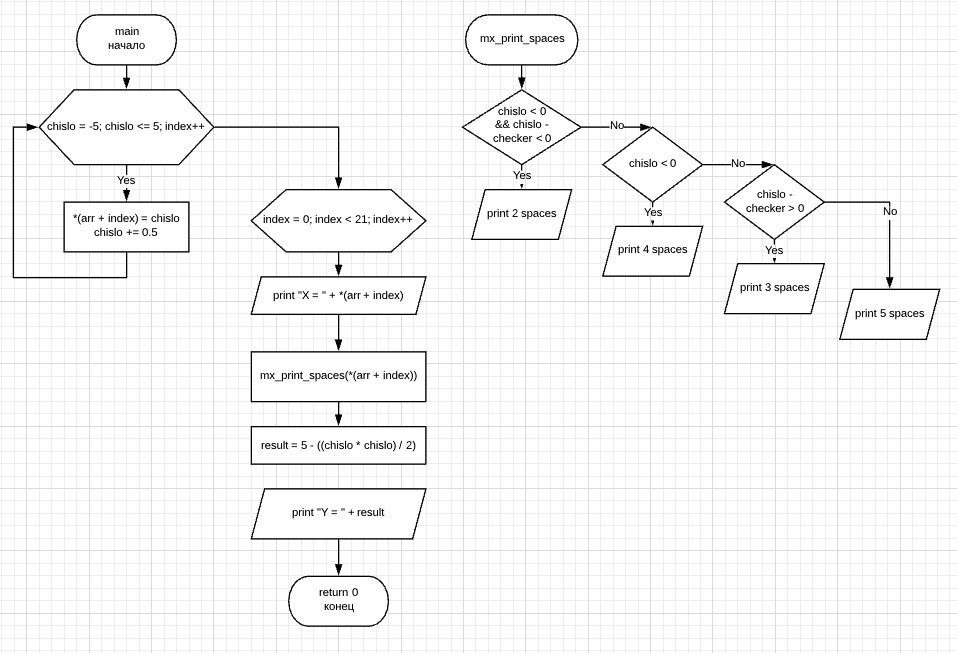
std::cout << "\nMin age is - " << min;

std::cout << "\nMax age is - " << max;

return 0;

}

# Составить таблицу значений функции y = 5 - x2/2 на отрезке [-5; 5] с шагом 0.5.



#include <iostream>

static double mx\_func\_result(double chislo) {

double result = 0;

result = 5 - ((chislo \* chislo) / 2);

return result;

}

static void mx\_print\_spaces(double chislo) {

int checker = chislo;

if(chislo < 0 && chislo - checker < 0)

std::cout << " ";

else if(chislo < 0)

std::cout << " ";

else if(chislo - checker > 0)

std::cout << " ";

else

std::cout << " ";

}

int main() {

double \*arr = (double\*)malloc(sizeof(double) \* 21);

int index = 0;

double chislo = -5;

int result;

for(; chislo <= 5; index++) {

\*(arr + index) = chislo;

chislo += 0.5;

}

for(index = 0; index < 21; index++) {

std::cout << "X = " << \*(arr + index);

mx\_print\_spaces(\*(arr + index));

result = mx\_func\_result(\*(arr + index));

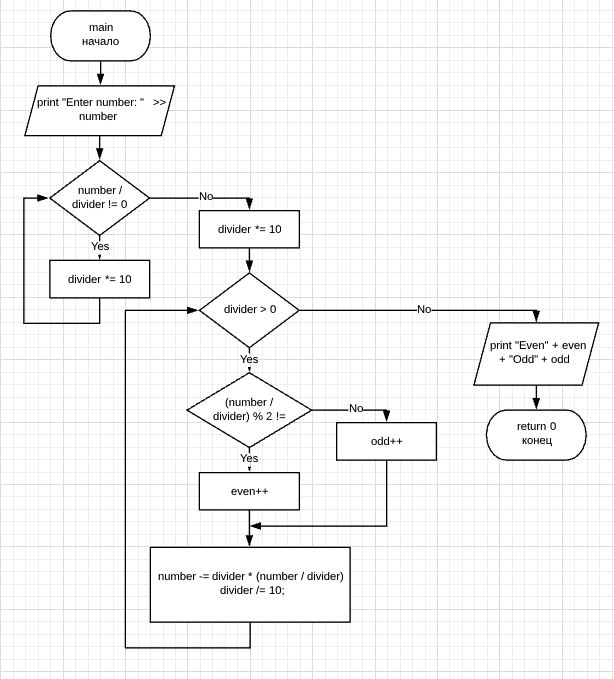
std::cout << "Y = " << result << "\n";

}

return 0;

}

# Написать программу, подсчитывающую количество четных и нечетных цифр в числе



#include <iostream>

int main() {

int odd = 0;

int even = 0;

int number;

int checker;

int divider = 10;

std::cout << "Enter number: ";

std::cin >> number;

while(number / divider != 0)

divider \*= 10;

divider /= 10;

while(divider > 0) {

if((number / divider) % 2 != 0)

even++;

else

odd++;

number -= divider \* (number / divider);

divider /= 10;

}

std::cout << "\nEven - " << even << ", Odd - " << odd;

return 0;

}