**РЕШЕНИ ПРИМЕРНИ ЗАДАЧИ**

**Задача c2:** Да се състави програма, с която да се въвеждат последователно от клавиатурата произволни числа, до въвеждането на стойност нула за край. Да се изчисли средноаритметичната стойност на положителните

въведени числа. Да се използва цикъл с постусловие (следусловие).

Средноаритметичната стойност се изчислява по формулата:

n

Σai

I=1

Srst= \_\_\_\_\_\_\_\_\_\_\_

n

**Решение:** За да се изчисли средно-

aритметичната стойност първо е

необходимо да се намери сумата и

броят на положителните числа,

които се въвеждат последователно

от клавиатурата. Това означава,

че всяко въведено число трябва да

се провери дали е положително и

само тогава да се добавя към

сумата и да се преброи.

За прекратяване на цикличните

действия по въвеждане и обработка

на числа служи въвеждането на

стойност нула.

Изчислението на средната стой-

ност може да се извърши само ако

е въведено поне едно положително

число, т.е. броят на положи-

телните числа **n** е различен от

нула, за да не се допуска

ситуация на опит за деление на 0.

**Използвани променливи величини:**

**srar** – средноаритметична стой-

ност;

**s** – сума на положителните числа;

**a** – променлива за стойността на

поредното въведено число;

**n** – брой на положителните числа.

**Програма на С:**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main(int argc, char \*argv[])**

**{ float a;** // *въвеждано число*

**double s=0, srar;** // *сума и средна стойност*

**int n=0;** // *брой на положителните числа*

**do** // *начало на циклична обработка*

**{ printf(" Въведи число: ");**

**scanf ("%f", &a);**

**if (a>0) { s += a;**

**n++; }**

**} while (a != 0);** // *край на цикъла*

**if (n != 0)**

**{ srar = s/n;**

**printf(" Средна стойност = %.2lf \n" , srar);**

**}**

**else**

**printf("Няма въведени положителни числа! \n");**

**system("PAUSE");**

**return 0;**

**}**

**Задача c3:** Да се състави програма, с която да се въвеждат последователно произволни числа от клавиатурата до въвеждането на стойност, равна на нула. Да се изчисли среднохармоничната стойност на въведените числа, като се използва формулата:

*Srh= \_\_\_n\_\_\_\_\_\_\_\_\_*

n

Σ 1/ai

I=1

където *n* е броят на числата;

*ai* - стойностите на числата.

**Решение:** За да се изчисли среднохармоничната стойност е

необходимо да се пресметне сумата от реципрочните стойности на въведените числа иброят на тези числа **n**.Реципрочна стойност на число

може да се изчисли, само ако това число е различно от нула.

В противен случай при изпълнението на програмата ще има за опит за деление на 0, и изпълнението на програмата ще се прекрати. Това налага

проверката за въведена стойност 0 да е преди пресмятането на реципрочната стойност на въведеното число, което на практика означава използване на цикъл с предусловие. Изчислението се извършва само ако има въведено поне едно число, различно от нула.

**Използват се следните променливи величини:**

**srh** – среднохармонична стойност;

**s** – сума от реципрочните стойности на въведените числа;

**a** – поредното въведено число;

**n** – брой на въведените числа.

**Програма на С:**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main(int argc, char \*argv[])**

**{ float a;** // *въвеждано число*

**double s=0, srh;** // *сума и средна стойност*

**int n=0;** // *брой на числата*

**printf(" Въведи число: ");**

**scanf ("%f", &a);**

**while (a != 0)** // *начало на циклична обработка*

**{ s += 1/a;**

**n++;**

**printf(" Въведи число: ");**

**scanf ("%f", &a);**

**}** // *край на цикъла*

**if (n != 0)** // *ако има въведени числа*

**{ srh = n/s;**

**printf("Средна хармонична = %.4lf \n", srh);**

**}**

**system("PAUSE");**

**return 0;**

**}**

**Задача c\_6:** Да се състави програма, която да изчислява стойността на функциятa

3 2

*y* = *ax* − (2*x* + 2) + 5

за всички цели стойности на **x** от 10 до 1 и произволно цяло число за ***a***.

Данните за **x** и **y** да се изведат в две колони.

**Решение:** Задача от този тип често се нарича "табулиране на функция". Според условието на задачата, изчисляването на стойността на функцията се изпълнява 10 пъти за стойности на **x** от 10 до 1. Това условие е подходящо, за да се използва цикъл с управляваща променлива, която намалява и да се използва **x** в качеството на управляваща променлива.

По условие всички данни са целочислени и аритметичните операции са с цели числа, резултатът от функцията също е цяло число.

При извеждането на резултатите, за да се подредят стойностите в две колони, се използват подходящи формати за извеждане и подходящ брой

празни позиции на екрана. При език С++ за форматиране се използват манипулатори от заглавен файл <**iomanip**>. За задаване на размер на полето е **setw**(), а действието му е само за едно извеждане след него.

**Използвани променливи:**

**a** – коефициент, въвежда се;

**x** – аргумент на функцията;

**y** – стойност на функциятa.

**Програма на С:**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main(int argc, char \*argv[])**

**{ int a, x, y;**

**printf ("Въведи a = ");**

**scanf ("%d", &a);**

**printf ("\n Табулиране на функция \n");**

**printf (" x y \n\n");**

**for (x=10; x>=1; x--)** // *цикъл с намаляващ брояч*

**{ y = a\*x\*x\*x - (2\*x+2)\*(2\*x+2) + 5;**

**printf ("%10d %10d \n", x, y);**

**}**

**system("PAUSE");**

**return 0;**

**}**

**Задачи за самостоятелна работа:**

1. Да се напише програма за намиране сумата на първите N естествени числа кратни на 3.

2. Да се напише програма за намиране произведението на първите N естествени числа (факториел N!).

3. Да се напише вариант на програмата c\_5 като се използва цикъл с предусловие за проверка на входните данни.

4. Да се напише програма, която намира следната сума:

*y* = cos *x* + cos2*x* + cos3*x* + ... + cos9*x*

при еднократно въведена стойност за ъгъла *x*. Да се направят варианти с използване на трите оператора за цикъл.

5. Да се напише програма за изчисляване на функцията *y* = 3sqrt( *x* + 2) + 3.9*x*2 , за всички цели стойности на *x* в интервала *x*∈[2,20].

6. Да се напише програма, която въвежда последователно от клавиатурата предварително неизвестен брой числа до въвеждането на стойност нула. Да се изчисли средноаритметичната стойност на тези от числата, които са в интервала (3,6).

7. Да се напише програма, която въвежда последователно от клавиатурата предварително неизвестен брой числа до въвеждането на стойност нула. Да се изчисли произведението на тези числа, които са кратни на 5.

**Съвети:**

Съобразете какъв тип данни е най-подходящо да се използва.

Можете да добавите и проверка за коректността на въведеното число

за N в задачи 1 и 2.

**Алгоритми и програми за работа с едномерни масиви**

**Задача d1:** Да се състави програма, която да намирасредната стойност на елементите на масив, който сесъстои от 6 елемента от реален тип.

**Решение:** Първо трябва да се въведат стойности за елементите на масива. След това всички елементи се сумират. Средната стойност се получава като се раздели получената сума на броя на елементите, който е 6.

При конкретно зададен брой за елементите на масива са възможни различни начини на работа. Единият начин е да се работи с константната стойност 6, а другият е да се дефинира именувана константа с тази стойност, например **n**=6. Вторият начин е по-добър, ако се налага програмата да се модифицира за друг брой елементи на масива. Тогава може да се промени само стойността на константата **n** и не се налага да се правят други промени в програмата. Обработката на елементите на масива е като се организира цикъл с управляваща променлива – брояч **i**, който се

променя за индексите от 1 до **n** (или от 1 до 6).

**Важно**: При масивите в език С и С++ първият елемент е с индекс 0. Затова вместо от 1 до 6, в програмата на С/С++ индексът **i** трябва да се изменя от 0 до 5, защото в паметта елемент с индекс 6 няма. Получава се разлика

между индекса на елемента в програмата и поредния му номер. Освен в програмата, това може се отрази и в извежданите за потребителя съобщения и резултати. За да се виждат на екрана поредни номера като 1, 2, 3... може вместо индекс **i** да се извежда стойността **i+1**.

Варианти на решение: В първия алгоритъм е реализиран един цикъл, в който са включени и въвеждането и сумирането на елементите. Този вариант изглежда по-кратък, но за програми с по-сложна обработка може да е неприложим. Вторият възможен вариант е с отделен цикъл за въвеждане на стойностите и отделен цикъл за обработка на масива за намиране на сумата на елементите му. Във втория алгоритъм, за брой на елементите на масива се използва именувана константа **n=6**.

**Използват се следните променливи:**

**t** – име на масива;

**s** – сума на елементите на масива;

**sr** – средна стойност;

**i** – управляваща променлива за цикъл и индекс на масива;

**n** – именувана константа за брой на елементите на масива.

**Програма на С** :

**#include <stdio.h>**

**#include <stdlib.h>**

**#define n 6**

/\* *дефинира се именувана константа* n=6 \*/

**int main(int argc, char \*argv[])**

**{ float t[n];** // *дефиниране на масив* t[6]

**int i;** // *индекс и брояч*

**double s, sr;**

**printf ("Въведи масив с %d елемента: \n", n);**

**for (i=0; i<n; i++)** // *цикъл с брояч за въвеждане*

**{ printf ("t[%d]= ", i+1);**

**scanf ("%f", &t[i]);** // *въвеждане на елемент*

**}**

**s=0.0;**

**for (i=0; i<n; i++)** // *цикъл с брояч за обработка*

**s += t[i];** // *сумиране на елементите*

**sr = s/n;**

**printf ("Средна стойност = %lf \n", sr);**

**system("PAUSE");**

**return 0;**

**}**

**Задача d2:** Да се въведат произволни реални стойности за елементите на масив CD, състоящ се от 13 елемента. Да се намери средната стойност на положителните елементи на масива. Да се изведат всички получени резултати.

**Решение:** За масива се въвеждат произволни стойности. При обработката е необходимо всички елементи на масива да се проверят дали отговарят на условието да са положителни. Задачата се свежда до намиране на сумата и

броя на положителните елементи на масива, от които може да се пресметне средната стойност. За да се избегне опит за деление на О, изчислението

за средна стойност е възможно само, ако има поне един положителен елемент. Това налага да се направи проверка дали има резултати и полученият брой е различен от 0.

**Използват се следните променливи:**

**cd** – едномерен масив;

**i** – управляваща променлива на цикъла и текущ индекс на

елемент от масива;

**br** – брой на положителните елементи;

**s** – сума на положителните елементи;

**sr** – средна стойност на положителните елементи.

**Програма на С++**

**#include <cstdlib>**

**#include <iostream>**

**using namespace std;**

**int main(int argc, char \*argv[])**

**{ float cd[13];** // *масив*

**int i, br;**

**double s, sr;**

**cout << "Въведи масив с 13 елемента: \n";**

**for (i=0; i<13; i++)** // *цикъл с брояч* i

**{ cout << "cd[" << i+1 << "]=";**

**cin >> cd[i];** // *въвеждане на елемент*

**}**

**s=0.0;**

**br=0;**

**for (i=0; i<13; i++)** // *цикъл с брояч* i

**if(cd[i]>0)**

**{ s += cd[i];** // *сумиране на елементите*

**br++;** // *преброяване +1*

**}**

**if (br!=0)** // *проверка има ли резултати*

**{ sr = s/br;**

**cout << "Сума на положителните = " << s << endl;**

**cout << "Брой на положителните = " << br << endl;**

**cout << "Средна стойност = " << sr << endl;**

**}**

**else**

**cout << "Няма положителни числа. \n";**

**system("PAUSE");**

**return EXIT\_SUCCESS;**

**}**

**Задача d3:** За едномерен масив **t**, който съдържа 10 реални елемента, да се намери стойността на максималния елемент и неговия пореден номер.

**Решение:** За решаването на тази задача трябва да се въведат две помощни променливи. Променливата **max** трябва да получи стойността на максималният елемент, а втората променлива **k** – индексът на този елемент от масива. Първо променливите **max** и **k** трябва да получат подходящи начални стойности, най-добре: стойността и индекса на първия елемент от масива. След това стойността **max** се сравнява последователно с всички елементи на масива. Ако сравняването открие елемент от масива, който е по-голям от текущата стойност на **max**, то тогава **max** получава нова стойност - стойността на този елемент, а променливата за индекса **k** – получава индекса му. Така когато се обработят всички елементи на масива,

стойността на **max** е равна на стойността на най-големия елемент на масива, а стойността на **k** е индекса му.

**Използвани променливи:**

**t** – масив;

**i** – текущ индекс за елемент на масива и управляваща променлива за цикъла;

**max** – максимален елемент;

**k** – индекс на максималния елемент.

**Програма на С:**

**#include <stdio.h>**

**int main(int argc, char \*argv[])**

**{ float t[10], max;**

**int i, k;**

**printf ("Въведи 10 числа: \n");**

**for (i=0; i<10; i++)** // *въвеждане на масива*

**{ printf ("t[%d]= ", i+1);**

**scanf ("%f", &t[i]);**

**}**

**max = t[0];** // *начални стойности за* max *и* k

**k = 0;**

**for (i=1; i<10; i++)** // *обработка на масива*

**if (t[i]>max)**

**{ max = t[i];** // *нови стойности за* max *и* k

**k = i;**

**}**

**printf (" Mаксимална стойност = %.2f \n", max);**

**printf (" Пореден номер = %d \n", k+1);**

**system("PAUSE");**

**return 0;**

**}**

**Задача е\_2:** Да се състави програма по следното условие:

Даден е двумерен масив **D(N,N)** с произволни въвеждани

реални стойности, като **N**<=50. Да се изведе матрицата в

табличен вид на екрана. Да се намери произведението на

елементите, които са по-големи от 10 и са разположени

над главния диагонал. Да се изведе получения резултат.

**Решение:** При въвеждането на стойност за брой редове и стълбове е необходимо да се направи проверка за коректни данни. За извършване на проверката може да се използва оператор за цикъл с предусловие или след-

условие. В случая е използван оператор със следусловие. Следва въвеждане на стойности за елементите на двумерния масив, а после – извеждането му в табличен вид. Обработката на масива над главния диагонал би могла да е с проверка за индексите да са **i<j**, или както е

реализирано в тази задача индексите **i** и **j** в цикличните структури се изменят по подходящ начин и така масивът се обработва само над главния диагонал. Елементите от матрицата се проверяват по зададеното условие. Извежда се полученото произведение.

**Използват се следните променливи:**

**d** – име на двумерен масив (матрица);

**n** – въвеждан брой на редове и стълбове;

**p** – променлива за произведението;

**i, j** – управляващи променливи за цикли и текущи индекси

на елемент от масива;

**Програма на С:**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main(int argc, char \*argv[])**

**{ int i, j, n;**

**float d[50][50];**

**double p;**

**do**

**{ printf("Въведи брой за ред и стълб (n<=50) n=");**

**scanf("%d",&n);**

**} while (n<2 || n>50);**

**printf("\n Въведи елементи на матрицата \n");**

**for (i=0; i<n; i++)** // *въвеждане на матрицата*

**for (j=0; j<n; j++)**

**{ printf("d[%d,%d]=", i+1, j+1);**

**scanf("%f", &d[i][j]);**

**}**

// *извеждане на матрицата в табличен вид*

**printf("\n\n\t Матрицата e: \n");**

**for (i=0; i<n; i++)**

**{ for (j=0; j<n; j++)**

**printf(" %10.2f ", d[i][j]);**

**printf("\n");**

**}**

**p=1;**

// *обработка на матрицата*

**for (i=0; i<n-1; i++)** // *обработка над гл.диагонал*

**for (j=i+1; j<n; j++)**

**if (d[i][j] > 10)**

**p \*= d[i][j];**

**printf("\n Произведението на елементите >10 над**

**главния диагонал p = %e \n", p);**

**system("PAUSE");**

**return 0;**

**}**