Міністерство освіти і науки України

Національний авіаційний університет

Кафедра комп'ютерних систем та мереж

**Звіт про виконання домашнього завдання №1**

**з дисципліни:**

"Програмування"

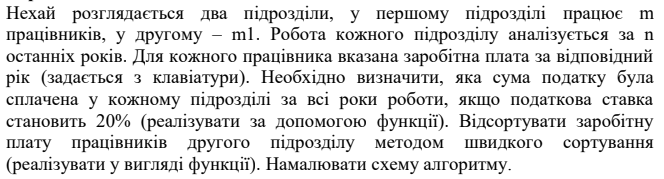
**Варіант 6**

Виконав:

Студент групи КС-133Б  
Курко М.О

Прийнла:  
Вавіленкова А.І.

Київ 2022



#define \_CRT\_SECURE\_NO\_WARNINGS

#include <Windows.h>

#include <stdio.h>

#include <malloc.h>

void quickSort(int\* arr, int size)

{

int i = 0;

int j = size - 1;

int mid = arr[size / 2];

do {

while (arr[i] < mid)

{

i++;

}

while (arr[j] > mid)

{

j--;

}

if (i <= j)

{

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

i++;

j--;

}

}

while (i <= j);

if (j > 0)

{

quickSort(arr, j + 1);

}

if (i < size)

{

quickSort(&arr[i], size - i);

}

}

int inputNumOfWorkers(int numOfSub)

{

printf("Введіть кількість працівників в %d компанії: ", numOfSub);

int temp = 0;

scanf("%d", &temp);

return temp;

}

int inputNumberOfYearsToAnalyze()

{

printf("За скільки останніх років проводиться аналіз?: ");

int temp = 0;

scanf("%d", &temp);

return temp;

}

void initArray(int \*\*arr, int weigth, int heigth)

{

for (int i = 0; i < weigth; i++)

{

arr[i] = malloc(sizeof(\*\*arr) \* heigth);

}

}

void fillingArray(int\*\* array, int weigth, int heigth)

{

for (int i = 0; i < weigth; i++)

{

for (int j = 0; j < heigth; j++)

{

printf("Введіть зарплату %d прцівника за %d рік: ", (i + 1), (j + 1));

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

}

printf("\n");

}

}

void outTwoDimensionalDynamiclArray(int\*\* array, int weigth, int heigth)

{

for (int i = 0; i < weigth; i++)

{

printf("Зарплатня %d працівника\n", (i + 1));

for (int j = 0; j < heigth; j++)

{

printf("%d\t", array[i][j]);

}

printf("\n");

}

}

void fillingAllData(int\*\* firstArr, int\*\* secondArr, int firstNumOfWorkers, int secondNumOfWorkers, int yearsForAnalys)

{

printf("Перший підрозділ:\n");

fillingArray(firstArr, firstNumOfWorkers, yearsForAnalys);

printf("Другий підрозділ:\n");

fillingArray(secondArr, secondNumOfWorkers, yearsForAnalys);

}

void sortSalaryInSubdivision(int \*\*arr, int heigth, int weigth)

{

for (int i = 0; i < weigth; i++)

{

int\* temp = 0;

temp = malloc(sizeof(\*temp) \* heigth);

for (int j = 0; j < heigth; j++)

{

temp[j] = arr[i][j];

}

quickSort(temp, heigth);

for (int j = 0; j < heigth; j++)

{

arr[i][j] = temp[j];

}

}

}

int sumSalaryInSubdivision(int\*\* arr, int heigth, int weigth)

{

int sum = 0;

int temp = 0;

for (int i = 0; i < weigth; i++)

{

for (int j = 0; j < heigth; j++)

{

temp = arr[i][j];

sum += temp;

}

}

return sum;

}

double sumOfSalaryAfterTax(int\*\* arr, int heigth, int weigth)

{

double sum = sumSalaryInSubdivision(arr, heigth, weigth);

double onePercent = sum / 100;

const taxPercent = 20;

double tax = onePercent \* taxPercent;

return sum -= tax;

}

void outAllInfoForSalary(int \*\*firstArr, int \*\*secondArr, int firstNumOfWorkers, int secondNumOfWorkers, int yearsForAnalys)

{

printf("Зарплатня до сортування\n");

printf("Перший підрозділ\n");

outTwoDimensionalDynamiclArray(firstArr, firstNumOfWorkers, yearsForAnalys);

printf("\n");

printf("Другий підрозділ\n");

outTwoDimensionalDynamiclArray(secondArr, secondNumOfWorkers, yearsForAnalys);

sortSalaryInSubdivision(secondArr, yearsForAnalys, secondNumOfWorkers);

printf("\n");

printf("Зарплатня після сортування сортування\n");

printf("Другий підрозділ\n");

outTwoDimensionalDynamiclArray(secondArr, secondNumOfWorkers, yearsForAnalys);

int firstSallaryNoTax = sumSalaryInSubdivision(firstArr, yearsForAnalys, firstNumOfWorkers);

int secondSallaryNoTax = sumSalaryInSubdivision(secondArr, yearsForAnalys, secondNumOfWorkers);

printf("\nСума зарплатні за всі роки без урахування податку\n");

printf("У першому підрозділі: %d\n", firstSallaryNoTax);

printf("У другому підрозділі: %d\n", secondSallaryNoTax);

double firstSallaryAfterTax = sumOfSalaryAfterTax(firstArr, yearsForAnalys, firstNumOfWorkers);

double secondSallaryAfterTax = sumOfSalaryAfterTax(secondArr, yearsForAnalys, secondNumOfWorkers);

printf("\nСума зарплатні за всі роки з урахуванням податку\n");

printf("У першому підрозділі: %lf\n", firstSallaryAfterTax);

printf("У другому підрозділі: %lf", secondSallaryAfterTax);

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

const int firstSub = 1;

const int secondSub = 2;

int numOfWorkersFirstSub = inputNumOfWorkers(firstSub);

int numOfWorkersSecondSub = inputNumOfWorkers(secondSub);

int numOfYears = inputNumberOfYearsToAnalyze();

int\*\* firstSubArr = 0;

firstSubArr = malloc(sizeof(\*firstSubArr) \* numOfWorkersFirstSub);

initArray(firstSubArr, numOfWorkersFirstSub, numOfYears);

int\*\* secondSubArr = 0;

secondSubArr = malloc(sizeof(\*secondSubArr) \* numOfWorkersSecondSub);

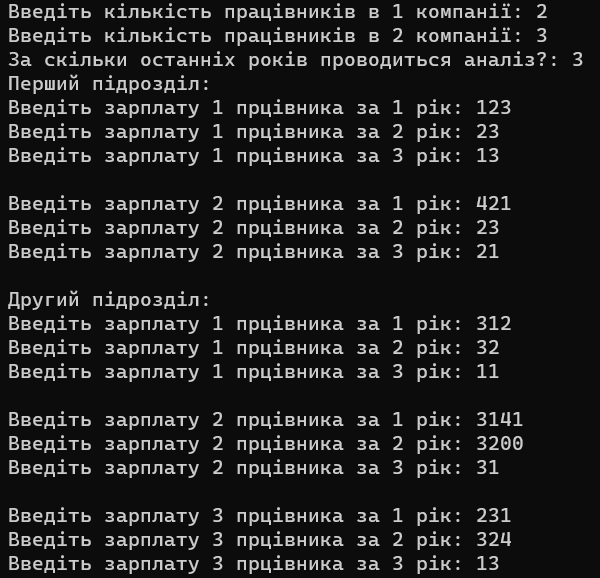
initArray(secondSubArr, numOfWorkersSecondSub, numOfYears);

fillingAllData(firstSubArr, secondSubArr, numOfWorkersFirstSub, numOfWorkersSecondSub, numOfYears);

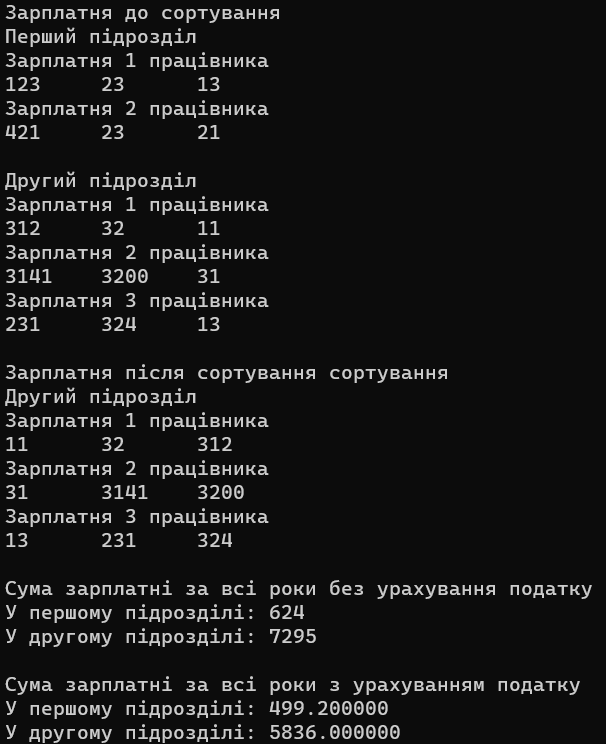
outAllInfoForSalary(firstSubArr, secondSubArr, numOfWorkersFirstSub, numOfWorkersSecondSub, numOfYears);

}

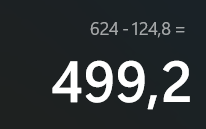
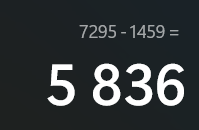
Введення інформації:



Вивід програми в консоль:



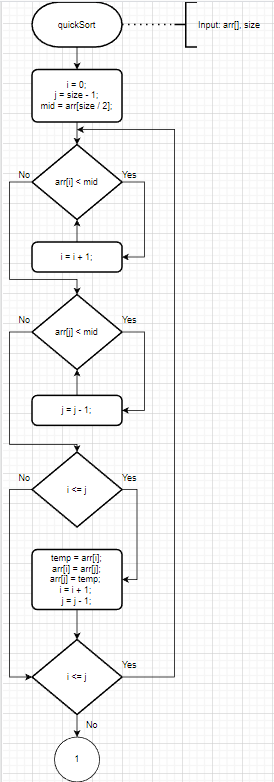
Перевіримо через калькулятор чи правильно рахується відсоток:

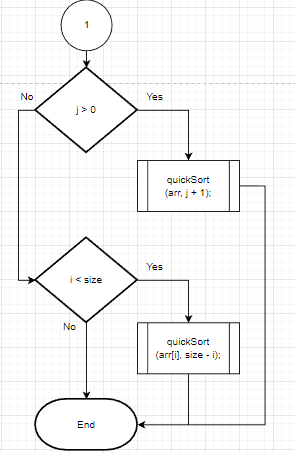


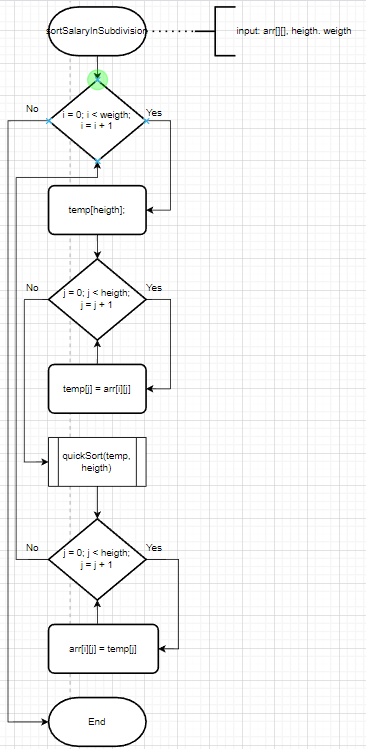
Судячи з результатів обрахунків можемо зробити висновок, що рахує вірно.

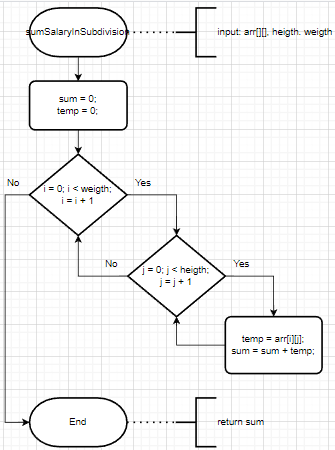
Можливі проблеми з переповненням пам’яті коли кількість працівників буде велика, але ця проблема не була виявлена при тестах, але все ж таки якщо буде така проблема потрібно просто очищати пам'ять після того як той чи інший масив нам більше не потрібен, але як і в МКР особливого сенсу від цього немає в нашому випадку.

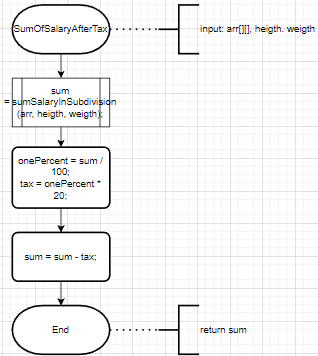
Блок-схеми:

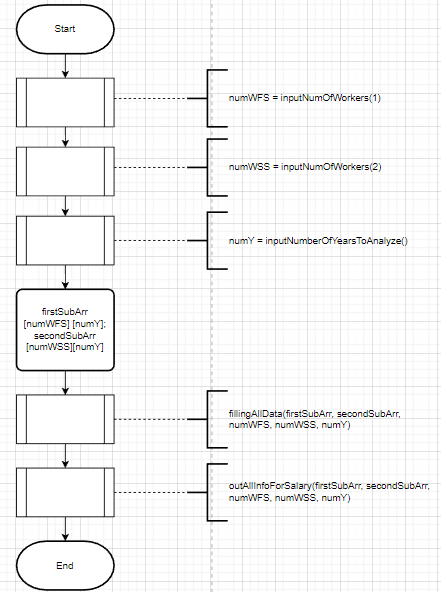












Не має сенсу описувати блок-схемами інші ф-ції, вони вже були описані в МКР або вся їх робота зводиться до виклику інших ф-цій і виведення інформації лінійно та без складних моментів.