

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение высшего образования  
МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ ИМ. К.Г.  
РАЗУМОВСКОГО (ПЕРВЫЙ КАЗАЧИЙ УНИВЕРСИТЕТ)  
(ФГБОУ ВО МГУТУ им. К.Г.РАЗУМОВСКОГО (ПКУ))



---

Факультет цифровых технологий

Кафедра: Информационные системы и цифровые технологии  
Направление подготовки – 09.03.01 «Проектирование и разработка  
программного обеспечения»

ОТЧЕТ ПО ДИСЦИПЛИНЕ

«Основы алгоритмизации и программирования»

Лабораторная работа № 7.

Вариант №3

Тема: «Задача со сложной структурой данных»

Преподаватель	Таченков О.С. <hr/> (ученая степень, ученое звание, фамилия, инициалы)
Студент	1 090301-РПРОо-24/1 Асылбек уулу Бакыт <hr/> курс группа (фамилия, имя, отчество)

Москва, 2024 г.

## **СОДЕРЖАНИЕ**

СЛОВЕСНАЯ ПОДСТАНОВКА ЗАДАЧИ.....	3
МАТЕМАТИЧЕСКАЯ ПОДСТАНОВКА ЗАДАЧИ.....	5
ЛИСТИНГ ПРОГРАММЫ.....	6
РЕЗУЛЬТАТЫ .....	9

## СЛОВЕСНАЯ ПОДСТАНОВКА ЗАДАЧИ

Составить выходную форму со всеми требуемыми выходными данными на основе введенных данных. Для программной организации работы с данными использовать структуры данных. Реализовать в программе:

- создание и заполнение структуры;
- вывод структуры;
- расчет данных по заданию.

По результатам обследования предприятий выяснить процентное выполнение плана по персоналу. Для каждого предприятия известно: наименование предприятия, фактическая численность персонала (промышленного и непрямошленного) и плановая численность всего персонала. Число предприятий не более 30. Результаты распечатать в виде таблицы:

Сведения о предприятиях, в которых выполнение плана по персоналу не менее... %

№ п/п	Наименование предприятия	Всего персонала по плану	Фактически персонала		Доля непрямошл енного персонала	Выполнение плана по персоналу
			промыш ленного	непрямош ленного		
1						
Итого:						

Указания: Заполнить исходную таблицу и сортировать ее по уменьшению значений второго столбца. После этого в цикле, пока пользователь не откажется

- запрашивать нижнюю границу процента выполнения плана по персоналу;
- копировать из исходной в рабочую таблицу строки с процентом выполнения плана по персоналу, большим заданного;
- выявлять предприятие с наименьшей долей непрямошленного персонала и запоминать его наименование;

- выдавать сведения о предприятиях.

## МАТЕМАТИЧЕСКАЯ ПОДСТАНОВКА ЗАДАЧИ

Дано:

*iae.name* – имя предприятия,

*iae.allEmployees* (A) – всего персонала по плану,

*iae.industrialEmployees* (IE) – кол-во промышленного персонала,

*iae.notIndustrialEmployees* (NIE) – кол-во непромышленного персонала,

*iae.notIndustrialEmployeesPercentage* (NIEP) – доля непромышленного персонала.

*iae.planForEmployees* (P) – план по персоналу.

*enterprises*[*iae*<sub>1</sub>, *iae*<sub>2</sub>, *iae*<sub>*i*</sub> ...], где *i* = 0 ... 30

$$NIEP = \frac{NIE}{A}, P = \frac{IE + NIE}{A}$$

Отображение нижней границы процента выполнения плана по персоналу:

$$iae[i].P < 50\%, \text{ где } i = 0 \dots n$$

Отображать записи с процентом выполнения плана по персоналу, большим заданного:

*iae.P* < *p1*, где *p1* заданный от пользователя процент выполнения плана по персоналу.

Поиск предприятия с наименьшей долей непромышленного персонала:

$$iae[i].NIEP < iae.NIEP, i = 0 \dots N$$

## ЛИСТИНГ ПРОГРАММЫ

```
#include <stdio.h>
#include <stdlib.h>

struct InformationAboutEnterprise {
    char* name;
    int allEmployees;
    int industrialEmployees;
    int notIndustrialEmployees;
    float notIndustrialEmployeesPercentage;
    float planForEmployees;
} typedef infAboutEnterprise;

infAboutEnterprise* creatingAndFillingIAEs(int n) {
    infAboutEnterprise* enterprises = calloc(n, sizeof(infAboutEnterprise));
    if (enterprises == NULL) {
        free(enterprises);
        return nullptr;
    }
    for (int i = 0; i < n; i++) {
        printf("Enterprise %d\n", i + 1);
        printf("name:");
        enterprises[i].name = calloc(20, sizeof(char));
        if (enterprises[i].name == NULL) {
            for (int j = 0; j <= i; j++) free(enterprises[i].name);
            free(enterprises);
            return nullptr;
        }
        scanf_s("%s", enterprises[i].name, 20);
        printf("allEmployees:");
        scanf_s("%d", &enterprises[i].allEmployees);
        printf("industrialEmployees:");
        scanf_s("%d", &enterprises[i].industrialEmployees);
        printf("notIndustrialEmployees:");
        scanf_s("%d", &enterprises[i].notIndustrialEmployees);
        enterprises[i].notIndustrialEmployeesPercentage =
            ((float)enterprises[i].notIndustrialEmployees /
            (float)enterprises[i].allEmployees) * 100;
        enterprises[i].planForEmployees =
            ((float)(enterprises[i].industrialEmployees +
            enterprises[i].notIndustrialEmployees) / (float)enterprises[i].allEmployees) *
            100;
    }
    return enterprises;
}

infAboutEnterprise* sortedIAEsByAllEmployees(int n, infAboutEnterprise*
enterprises) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (enterprises[i].allEmployees < enterprises[j].allEmployees) {
                infAboutEnterprise temp = enterprises[i];
                enterprises[i] = enterprises[j];
                enterprises[j] = temp;
            }
        }
    }
    return enterprises;
}

void printIAETableHeader() {
    printf("+---+---+---+---+---+---+---+---+\n");
    printf("| Num |   Name   | AE  | IE  | NIE  | NIE (%) | PFE | \n");
```



```

        printLowerBoundOfPlanForEmployees(n, enterprises);
        break;
    case 2:
        printIAEFilteredByPlanForEmployees(n, enterprises);
        break;
    case 3:
        printf("%s\n", enterpriseName);
        break;
    case 4:
        printIAETableHeader();
        int totalEmployees = 0, totalIndustrialEmployees = 0,
totalNotIndustrialEmployees = 0;
        for (int i = 0; i < n; i++) {
            printIAE(enterprises[i], i);
            totalEmployees += enterprises[i].allEmployees;
            totalIndustrialEmployees +=
enterprises[i].industrialEmployees;
            totalNotIndustrialEmployees +=
enterprises[i].notIndustrialEmployees;
        }
        float totalNotIndustrialEmployeesPercentage =
((float)totalNotIndustrialEmployees / (float)totalEmployees) * 100;
        float totalPlanForEmployees = ((float)(totalIndustrialEmployees
+ totalNotIndustrialEmployees) / (float)totalEmployees) * 100;
        printf("\n\n");
        printf("| total:\t\t|\n");
        printf("| %d ", totalEmployees);
        printf("| %d ", totalIndustrialEmployees);
        printf("| %d ", totalNotIndustrialEmployees);
        printf("| %.0f ", totalNotIndustrialEmployeesPercentage);
        printf("| %.0f |\n", totalPlanForEmployees);
        break;
    default:
        printf("bye bye!");
        break;
    }
}
if (enterpriseName != NULL) free(enterpriseName);
for (int i = 0; i < n; i++) {
    if (enterprises[i].name != NULL) free(enterprises[i].name);
}
free(enterprises);
return 0;
}

```



## РЕЗУЛЬТАТЫ

```
input the number of enterprises:3
Enterprise 1
name:company_X
allEmployees:44500
industrialEmployees:14500
notIndustrialEmployees:6000
Enterprise 2
name:company_C
allEmployees:40000
industrialEmployees:10000
notIndustrialEmployees:9999
Enterprise 3
name:company_Z
allEmployees:55000
industrialEmployees:35000
notIndustrialEmployees:9000

+-----+
| Num |   Name   | AE  | IE  | NIE | NIE (%) | PFE |
+-----+
| 1   | company_Z | 55000 | 35000 | 9000 | 16      | 80   |
+-----+
| 2   | company_X | 44500 | 14500 | 6000 | 13      | 46   |
+-----+
| 3   | company_C | 40000 | 10000 | 9999 | 25      | 50   |
+-----+
```

```
1 - lowerBoundOfPlanForEmployees
2 - printIAEFilteredByPlanForEmployees
3 - smallestNotIndustrialEmployeesPercentageEnterprise
4 - print all
0 - exit
input i:1

+-----+
| Num |   Name   | AE  | IE  | NIE | NIE (%) | PFE |
+-----+
| 2   | company_X | 44500 | 14500 | 6000 | 13      | 46   |
+-----+
| 3   | company_C | 40000 | 10000 | 9999 | 25      | 50   |
+-----+

1 - lowerBoundOfPlanForEmployees
2 - printIAEFilteredByPlanForEmployees
3 - smallestNotIndustrialEmployeesPercentageEnterprise
4 - print all
0 - exit
input i:
```

```

1 - lowerBoundOfPlanForEmployees
2 - printIAEFilteredByPlanForEmployees
3 - smallestNotIndustrialEmployeesPercentageEnterprise
4 - print all
0 - exit
input i:2
input planForEmployees:49
+-----+-----+-----+-----+-----+-----+
| Num |   Name   |  AE  |  IE  |  NIE  |  NIE (%) | PFE |
+-----+-----+-----+-----+-----+-----+
| 1   | company_Z | 55000 | 35000 | 9000 |    16    | 80 |
+-----+-----+-----+-----+-----+-----+
| 3   | company_C | 40000 | 10000 | 9999 |    25    | 50 |
+-----+-----+-----+-----+-----+-----+
1 - lowerBoundOfPlanForEmployees
2 - printIAEFilteredByPlanForEmployees
3 - smallestNotIndustrialEmployeesPercentageEnterprise
4 - print all
0 - exit
input i:

```

```

1 - lowerBoundOfPlanForEmployees
2 - printIAEFilteredByPlanForEmployees
3 - smallestNotIndustrialEmployeesPercentageEnterprise
4 - print all
0 - exit
input i:3
company_X
1 - lowerBoundOfPlanForEmployees
2 - printIAEFilteredByPlanForEmployees
3 - smallestNotIndustrialEmployeesPercentageEnterprise
4 - print all
0 - exit
input i:4
+-----+-----+-----+-----+-----+-----+
| Num |   Name   |  AE  |  IE  |  NIE  |  NIE (%) | PFE |
+-----+-----+-----+-----+-----+-----+
| 1   | company_Z | 55000 | 35000 | 9000 |    16    | 80 |
+-----+-----+-----+-----+-----+-----+
| 2   | company_X | 44500 | 14500 | 6000 |    13    | 46 |
+-----+-----+-----+-----+-----+-----+
| 3   | company_C | 40000 | 10000 | 9999 |    25    | 50 |
+-----+-----+-----+-----+-----+-----+

| total:           |
| 139500 | 59500 | 24999 | 18 | 61 |

```

```
1 - lowerBoundOfPlanForEmployees
2 - printIAEFilteredByPlanForEmployees
3 - smallestNotIndustrialEmployeesPercentageEnterprise
4 - print all
0 - exit
input i:0
bye bye!
```

```
input the number of enterprises:-25
error: n <= 0
Process finished with exit code 0
```

```
input the number of enterprises:0
error: n <= 0
Process finished with exit code 0
```