

ADDIS ABABA SCIENCE AND TECHNOLOGY UNIVERSITY

***DEPARTMENT OF SOFTWARE ENGINEERING***

Fundamentals of Programming II

**Section: E (Group V)**

|  |  |
| --- | --- |
| Name | ID |
| Mieraf Jejew | ETS0467/12 |
| Mikeyas Alemu | ETS0471/12 |
| Nahom Getachew | ETS0500/12 |
| Nahom Habatmu | ETS0505/12 |
| Namus Shukra | ETS0510/12 |

SUBMITTED TO: Mr. Chere Lemma

DATE OF SUBMISSION: Sep 23, 2021

**Program analysis for sorting function**

Function declaration

void sortEmpInfo(Employee employee[]);

Function Definition

* Input

➢ employee[]

* Output

➢ send array which hold indexes ordered by name and number which tell what type of sort it is

* Process/Operations

➢ variable declaration

1. int\* Index=new int[employeeCounter]; //used to hold sorted index and declared globally

to do not declare it again and not to lose the data that found on it

1. char\* names[employeeCounter]; //hold the full name of the employee used to sort name
2. char\* Min= new char ('|') //used to hold the minimum name while we compare the names
3. string fullname; //length used to hold size of full name
4. int temp,length,select; //temp used to hold index while comparison

// select used to choose menu

//Full name used to hold full name temporarily

* First of the function copy the name of the employees to the temporary dynamic array( char\* names[employeeCounter]) using for loop

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

{

length=sizeof(employee[i].fullName.firstName+employee[i].fullName.middleName+

employee[i].fullName.lastName);

names[i] = new char[length + 1];

fullname=(employee[i].fullName.firstName+' '+employee[i].fullName.middleName+' '+employee[i].fullName.lastName);

fullname.copy(names[i],length);

}

* Then compare every name to get ascending order, after the inner loop finished the iteration we will store the index of minimum name on array(int\* Index=new int[employeeCounter])

Then we will change the first character of the name to a character which has a higher ASCII code value to make it last in the next comparison. And also change the temporary name holder(MIn) to a character which has a higher ASCII code value but one step lower from the name array first letter.

for(int k=0; k<employeeCounter; k++) {

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

if(strcmp(Min,names[i])>=0) {

strcpy(Min,names[i]);

temp=i;

}

}

Min[0]='}';

Index[k]=temp;

names[temp][0]='|';

}

* Delete the temporary name holder

if(employeeCounter!=holdCounter)

{

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

{

delete [] names[i];

}

delete Min;

holdCounter=employeeCounter;

* Choose sort type by name or by ID

select = arrowMenu(sortMenu, 3)

* There is last number of employees holder which used , not to do comparison again if there is no change on the number of number of employees

**Program analysis for the employee 1information, former employee and pension employees**

Input

\*As all the functions are used for displaying, they received the data from the inputted information from the users or from void insertEmployeeRecord() function.

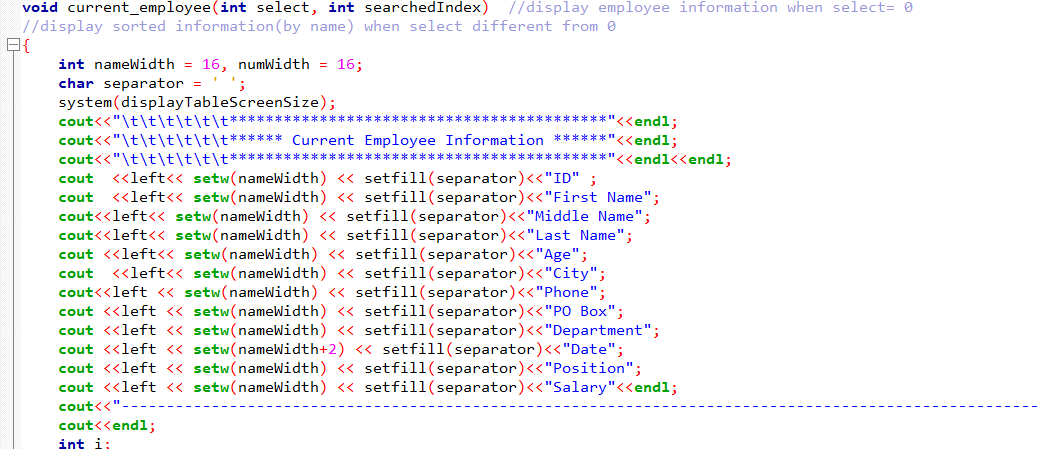
Output

* Id
* First name
* Middle name
* Last name
* Age
* City
* Phone
* Po Box
* Department
* Date
* Position
* pension
* Salary
* Tax
* Over time
* Gross pay
* Others
* Net pay

1.3 Process/Operations

* Initializing name width to 16 numwidth 16
* Initializing char separator
* Manipulating the screen size
* Displaying Id
* Displaying First name
* Displaying Middle name
* Displaying Last name
* Displaying Age
* Displaying City
* Displaying Phone
* Displaying Po Box
* Displaying Department
* Displaying Date
* Displaying Position
* Displaying Salary

2. Design of the function



The display functions have similar implementation. So it has been decided to include the above and to omit the others.

**Program Analysis for Employee statistical report function and financial report function**

1. **For statistical report**

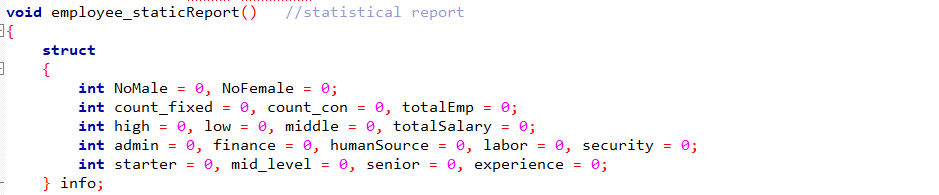
**Input:**

The function takes the input of the int main function and the constant that are declared above. Also receives from the sorting (manipulation) function.

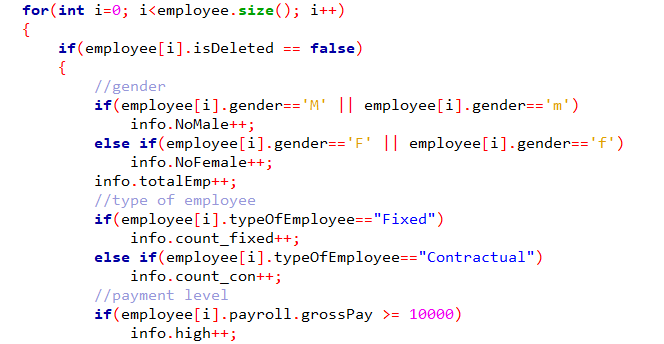
**Output:**

The function display Employee statistical reports in number.  
Most reports are output using counting the information input int main. And the it display the it.

**Operation:**

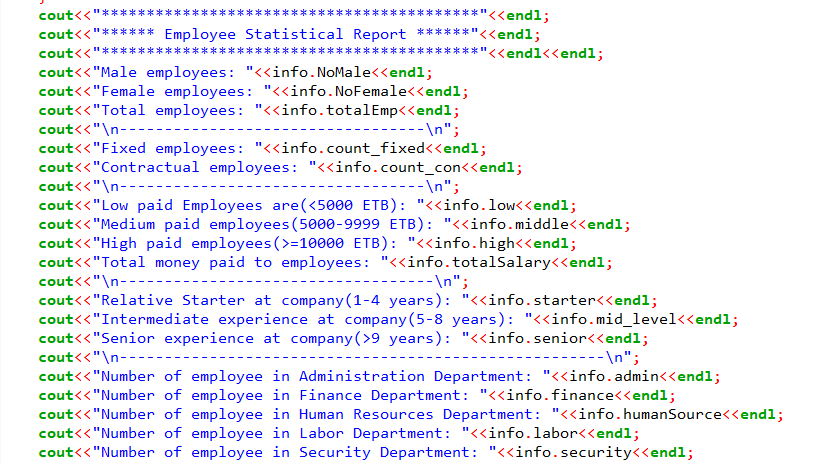
We divided each levels and give a variable like ‘high’, ‘low’, ‘middle’. And initialize them 0. So as the employee number increases in for loop it count starting from zero.****

We use a specific structure for the employee\_staticReport only. This make is for the counting the details from the main function.



After that we create if conditions in a for loop. The if condition constraint specifically what we want to count and give the information.

The for loop: as there is more than one employee we have to use the for loop that can make iteration for counting.



Finally we display using cout syntax. We call the iterations that are counted in for loop we created before.

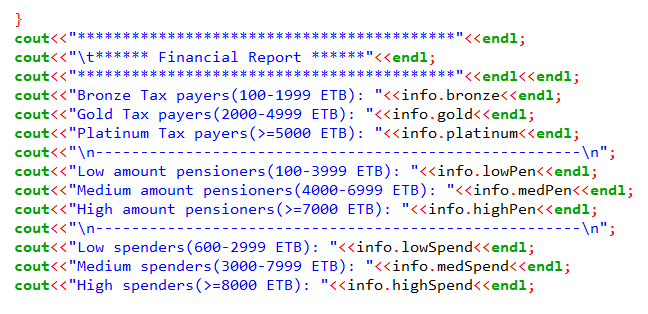
1. **For financial report**

**Input:**

The function takes the input of the int main function and the constant that are declared above. Also receives from the sorting (manipulation) function.

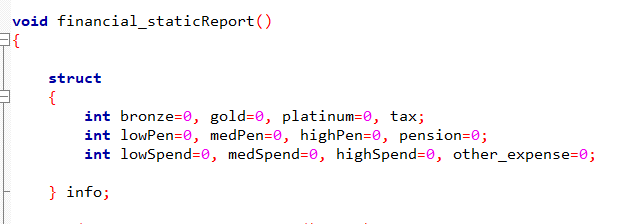
**Output:**

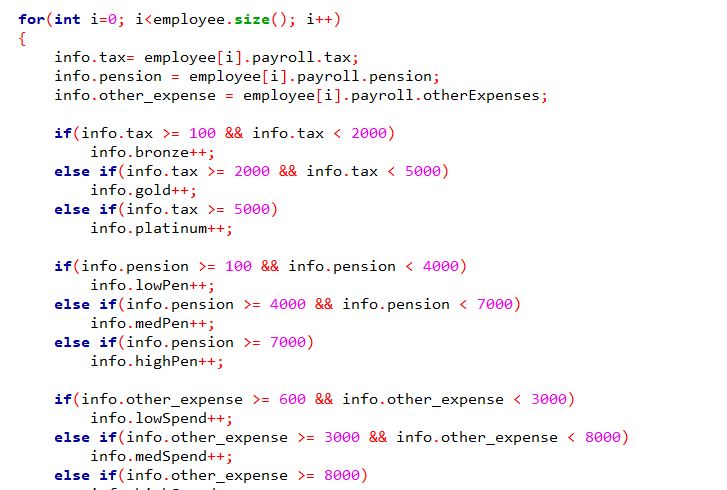
Again here we display using the cout syntax.

It display the counted data that are input in the main and counted in for loop the we will see in operation.

**Operation:**

The method we use here is almost the same to the employee statistical report. The difference here is the variables we use and the if conditions we use.





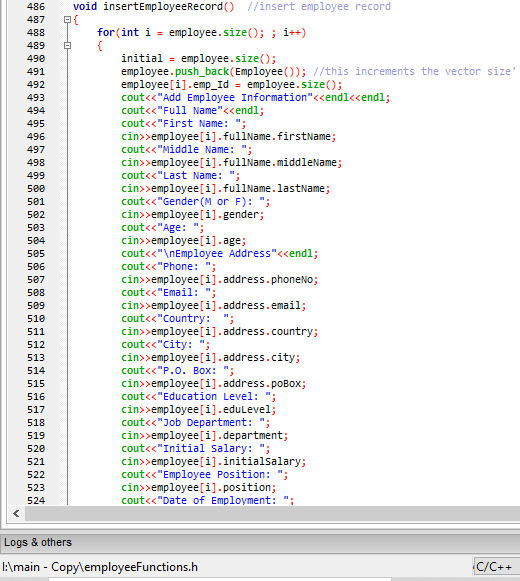
As we can see the algorithms we use are the same but for the conditions are different for the task given specifically.

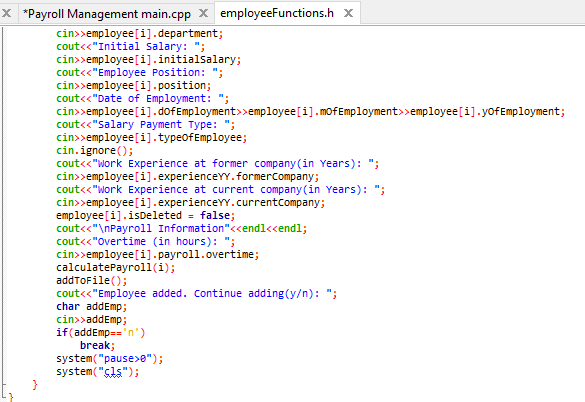
The same also for the structures.

**Program Analysis for the Record Management**

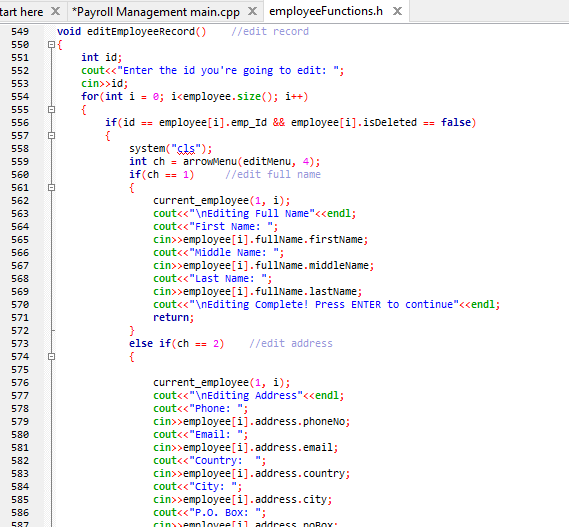
1. Input: all the inputs in the structure
2. Output: request to add required information
3. Operation

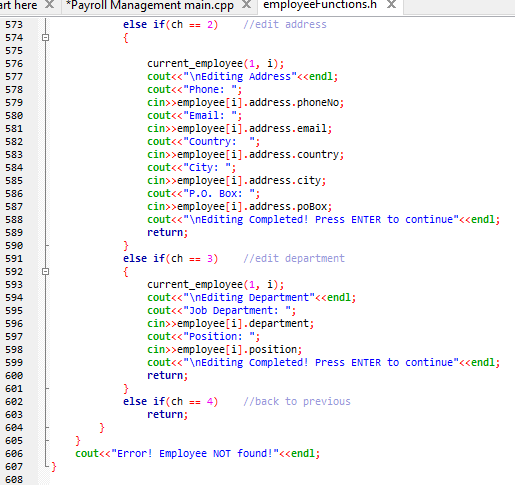
* String declaration
* Reading input data
* Adding new employee data
* Editing employee data
* Deleting employee data
* Printing the output



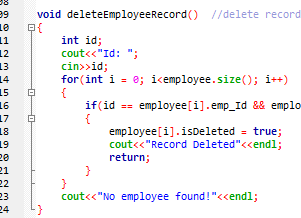


**Edit**

****

****

**Delete**



**Program Analysis for the search function**

1. Input: First name, Middle name, Last name, ID number;
2. Output

Printing EMPLOYEE FOUND or EMPLOYEE NOT FOUND.

1. Operation

* String declaration
* Function definition and Reading input data
* function declaration which searches employee using Id number
* Function definition and Reading input data
* Checking the employee using input data searchbyName
* Checking the employee using input data searchbyId
* Printing the output

