

```

1  #include <iostream>
2  #include <windows.h>
3  #include <fstream> // inclusão da biblioteca para arquivos
4
5  using namespace std;
6
7  struct population // declara o tipo population
8  {
9      float wage; // declara o tipo de cada membro
10     int age;
11     int numBofKids;
12     char sex;
13 };
14
15 float sumWage, medianWage, biggestWage, percentWomanWage; // variáveis para realizar os cálculos necessários
16 int sumKids, medianKids; // variáveis para realizar os cálculos necessários
17
18 const int amountPopulation = 10; // constante que guarda o número de habitantes da cidade, utilizada durante o código para determinar repetições e cálculos de
19 // média
20 population person[amountPopulation]; // armazena os dados no struct
21
22 ofstream write; // (tipo) ofstream escrita (variável acesso à funções de manipulação de arquivo);
23 ifstream read; // (tipo) ifstream leitura (variável acesso à funções de manipulação de arquivo);
24
25 void separator() // divisor de organização
26 {
27     cout << endl
28         << endl
29         << "»" << endl
30         << endl;
31 }
32
33 void insertData() // usuário insere manualmente os dados dos habitantes através de uma estrutura de repetição
34 {
35     for (int i = 0; i < amountPopulation; i++)
36     {
37         cout << "Digite o salário do " << i + 1 << "º habitante: ";
38         cin >> person[i].wage;
39
40         cout << "Digite a idade do " << i + 1 << "º habitante: ";
41         cin >> person[i].age;
42
43         cout << "Digite o número de filhos do " << i + 1 << "º habitante: ";
44         cin >> person[i].numBofKids;
45
46         cout << "Digite o sexo do " << i + 1 << "º habitante: ";
47         cin >> person[i].sex;
48
49         separator();
50         system("cls");
51     }
52 }
53
54 void calculateReport() // realiza os cálculos
55 {
56     for (int i = 0; i < amountPopulation; i++)
57     {
58         sumWage += person[i].wage; // soma o salário de todos os habitantes
59         sumKids += person[i].numBofKids; // soma o número de filhos de todos os habitantes
60         if (person[i].wage > biggestWage) // verifica e atribui o maior salário à variável biggestWage
61         {
62             biggestWage = person[i].wage;
63         }
64
65         tolower(person[i].sex); // converte o caractere em MAIÚSCULO
66
67         if (person[i].sex == 'F' && person[i].wage > 1000) // verifica se o habitante da posição é uma mulher e se recebe mais de R$1000,00, se sim incrementa 1 à
68         // contagem
69         {
70             percentWomanWage++;
71         }
72
73         medianWage = sumWage / amountPopulation; // calcula a média de salário
74         medianKids = sumKids / amountPopulation; // calcula a média de filhos
75         percentWomanWage = (percentWomanWage * 100) / amountPopulation; // calcula o percentual de mulheres que ganham mais de R$1000,00
76     }
77 }
78
79 void showReport() // mostra o relatório ao usuário
80 {
81     cout << "Média salarial dos habitantes da cidade: R$" << medianWage << endl;
82     cout << "Média de filhos dos habitantes da cidade: " << medianKids << endl;
83     cout << "O maior salário registrado na cidade: R$" << biggestWage << endl;
84     cout << "Percentual de mulheres com salário superior a R$1000,00: " << percentWomanWage << "%" << endl;
85 }

```

```

86 void saveReport() // salva o relatório no .txt
87 {
88     write.open("Report.txt");
89     write.clear();
90     write << "Média salarial dos habitantes da cidade: R$" << medianWage << endl;
91     write << "Média de filhos dos habitantes da cidade: " << medianKids << endl;
92     write << "O maior salário registrado na cidade: R$" << biggestWage << endl;
93     write << "Percentual de mulheres com salário superior a R$1000,00: " << percentWomanWage << "%" << endl;
94     write.close();
95 }
96
97 void saveData() // salva os dados no .txt
98 {
99     write.open("DataList.txt");
100    write.clear();
101    for (int i = 0; i < amountPopulation; i++)
102    {
103        write << person[i].wage << endl;
104        write << person[i].age << endl;
105        write << person[i].numbOfKids << endl;
106        write << person[i].sex << endl;
107    }
108    write.close();
109 }
110
111 void loadData() // carrega os dados do arquivo
112 {
113     read.open("DataList.txt"); // abre o arquivo ou cria um, se não existir
114     if (read.fail())           // se falhar a leitura do arquivo
115     {
116         cout << "ERROR!";
117         system("pause");
118         read.clear(); // fecha o arquivo
119     }
120
121     while (!read.eof())
122     {
123         for (int i = 0; i < amountPopulation; i++)
124         {
125             read >> person[i].wage;
126             read >> person[i].age;
127             read >> person[i].numbOfKids;
128             read >> person[i].sex;
129         }
130     }
131     read.close(); // fecha depois de carregar todos os arquivos
132 }
133
134 int main()
135 {
136     UINT CPAGE_UTF8 = 65001;
137     UINT CPAGE_DEFAULT = GetConsoleOutputCP();
138     SetConsoleOutputCP(CPAGE_UTF8);
139
140     int option = -1, saveOption = -1;
141
142     do // responsável pelo menu, onde o usuário insere a opção desejada
143     {
144         separator();
145         cout << "* |\t(1) - Inserir dados manualmente" << endl // mostra o menu ao usuário para que insira a opção desejada
146              << "* |\t(2) - Carregar dados de um arquivo" << endl
147              << "* |\t(0) - Sair" << endl
148              << "* |\tDigite uma opção: ";
149         cin >> option;
150         system("cls");
151
152         switch (option) // cada caso responde à opção do menu (Option)
153         {
154             case 0: //
155                 cout << "Saindo ... ";
156                 Sleep(1000);
157                 system("cls");
158                 break;
159
160             case 1:
161                 insertData(); // insere os dados manualmente
162                 calculateReport(); // realiza os cálculos para gerar o relatório
163                 separator();
164                 showReport(); // mostra o relatório
165                 separator();
166

```

```

167     do
168     {
169         cout << "\tDeseja salvar as informações?" << endl // imprime menu de salvar ou não os dados e relatório nos .txt
170         << "\t(1) - Sim" << endl
171         << "\t(2) - Não" << endl
172         << "\tDigite uma opção: ";
173         cin >> saveOption;
174
175         system("cls");
176
177         if (saveOption == 1) // salva
178         {
179             saveReport();
180             cout << "Relatório salvo!";
181             saveData();
182             cout << "Dados salvos!";
183         }
184         else if (saveOption == 2) // encerra o programa
185         {
186             cout << "Encerrando programa ... ";
187             Sleep(2000);
188             break;
189         }
190         else if (saveOption != 1 && saveOption != 2)
191         {
192             cout << "ERROR! Opção inválida ... ";
193             system("pause");
194         }
195     } while (saveOption != 2);
196
197     break;
198
199     default: // tratamento para valores inválidos no menu
200     {
201         system("cls");
202         cout << "ERROR! Opção inválida ... ";
203         system("pause");
204         break;
205     }
206 } while (option != 0);

```

```

207
208     cout << endl
209     << endl;
210     return 0; // fim do programa
211 }
212

```