

José Nelson

## Lab 8 – AP1

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>
4 #include <locale.h>
5
6 int main(void) {
7     int op;
8     setlocale(LC_ALL, "Portuguese_Brazil.1252");
9
10    do {
11        printf("\n----- MENU PRINCIPAL ----- \n");
12        printf("1 - Exercício 1 (Info Menu - já implementado pela estrutura)\n");
13        printf("2 - Exercício 2 (Valores de 100 a 1)\n");
14        printf("3 - Exercício 3 (Ímpares 101-201)\n");
15        printf("4 - Exercício 4 (Soma Pares Mult. 5, 1-500)\n");
16        printf("5 - Exercício 5 (Meio Triângulo de Asteriscos)\n");
17        printf("6 - Exercício 6 (Alturas de 5 Pessoas)\n");
18        printf("7 - Exercício 7 (Progressão Geométrica)\n");
19        printf("8 - Exercício 8 (Somatório Eq. 2)\n");
20        printf("9 - Exercício 9 (Série de Fibonacci)\n");
21        printf("10 - Exercício 10 (Rescreva 1019 - Quadrado Pares)\n");
22        printf("Digite -1 para SAIR\n");
23        printf("===== \n");
24        printf("Escolha uma opção: ");
25
26        if (scanf("%d", &op) != 1) {
27            printf("Entrada inválida. Por favor, digite um número.\n");
28            while (getchar() != '\n');
29            op = 0;
30            continue;
31        }
32
33        while (getchar() != '\n');
34
35        switch (op) {
36            case 1: {
37                printf("\n--- Exercício 1: Informação do Menu --- \n");
38                printf("Você está utilizando o menu iterativo solicitado.\n");
39                printf("O menu permite escolher os exercícios da lista e encerra com a opção -1. [cite: 1]\n");
40                break;
41            }
42            case 2: {
43                printf("\n--- Exercício 2: Valores de 100 a 1 --- \n");
44                for (int val = 100; val >= 1; val--) {
45                    printf("%d ", val);
46                }
47                printf("\n");
48            }
49        }
50    } while (op != -1);
51 }
```

```
50 }
51
52 case 3: {
53     printf("\n--- Exercício 3: Números ímpares entre 101 e 201 --- \n");
54     for (int val = 101; val <= 201; val += 2) {
55         printf("%d ", val);
56     }
57     printf("\n");
58     break;
59 }
60 case 4: {
61     printf("\n--- Exercício 4: Soma de pares múltiplos de 5 (1 a 500) --- \n");
62     int soma_ex4 = 0;
63     for (int i = 1; i <= 500; i++) {
64         if (i % 2 == 0 && i % 5 == 0) {
65             soma_ex4 += i;
66         }
67     }
68     printf("A soma dos números pares múltiplos de 5 entre 1 e 500 é: %d\n", soma_ex4);
69     break;
70 }
71 case 5: {
72     printf("\n--- Exercício 5: Meio Triângulo de Asteriscos --- \n");
73     int n_tri;
74     printf("Digite um valor N positivo para o triângulo: ");
75     if (scanf("%d", &n_tri) != 1) {
76         printf("Entrada inválida.\n");
77         while (getchar() != '\n');
78         break;
79     }
80     while (getchar() != '\n');
81     if (n_tri <= 0) {
82         printf("N deve ser um valor positivo.\n");
83     } else {
84         for (int i = 1; i <= n_tri; i++) {
85             for (int j = 1; j <= i; j++) {
86                 printf("*");
87             }
88             printf("\n");
89         }
90     }
91     break;
92 }
93 case 6: {
```

```
File Edit Selection View Go Run Terminal Help
Algoritmos de programação

int main(void) {
    do {
        switch (op) {
            case 5: {
                printf("\n--- Exercício 6: Alturas de 5 Pessoas ---\n");
                float altura_ex8, menor_alt = 0.0f, maior_alt = 0.0f, soma_alt = 0.0f;
                int alturas_validas = 0;

                for (int i = 1; i <= 5; i++) {
                    printf("Digite a altura da %dª pessoa (em metros, ex: 1.75): ", i);
                    if (scanf("%f", &altura_ex8) != 1 || altura_ex8 <= 0) {
                        printf("Altura Invalida. Por favor, insira um valor positivo.\n");
                        while (getchar() != '\n');
                        i--;
                        continue;
                    }
                    while (getchar() != '\n');
                    soma_alt += altura_ex8;
                    alturas_validas++;

                    if (alturas_validas == 1) {
                        menor_alt = altura_ex8;
                        maior_alt = altura_ex8;
                    } else {
                        if (altura_ex8 < menor_alt) {
                            menor_alt = altura_ex8;
                        }
                        if (altura_ex8 > maior_alt) {
                            maior_alt = altura_ex8;
                        }
                    }
                }

                if (alturas_validas == 5) {
                    float media_alt = soma_alt / 5;
                    printf("\nA menor altura do grupo e: %.2f\n", menor_alt);
                    printf("\nA maior altura do grupo e: %.2f\n", maior_alt);
                    printf("\nA altura media do grupo e: %.2f\n", media_alt);
                } else {
                    printf("\nNao foi possivel calcular os resultados pois nem todas as alturas foram inseridas corretamente.\n");
                }
                break;
            }
            case 7: {
                printf("\n--- Exercício 7: Progressao Geometrica ---\n");
                double a1_pg, q_pg, an_pg;
                int n_termos_pg;

                printf("Digite o valor inicial (a1) da PG: ");
                if (scanf("%lf", &a1_pg) != 1) {
                    printf("Entrada Invalida para a1.\n");
                    while (getchar() != '\n'); break;
                }
                while (getchar() != '\n');

                printf("Digite o valor da razao (q) da PG: ");
                if (scanf("%lf", &q_pg) != 1) {
                    printf("Entrada Invalida para q.\n");
                    while (getchar() != '\n'); break;
                }
                while (getchar() != '\n');

                printf("Digite o numero de termos (N) da PG: ");
                if (scanf("%d", &n_termos_pg) != 1 || n_termos_pg <= 0) {
                    printf("Numero de termos Invalido. Deve ser um inteiro positivo.\n");
                    if (n_termos_pg <= 0 && getchar() == '\n') {
                        while (getchar() != '\n');
                    } else while (getchar() != '\n');
                    break;
                }
                while (getchar() != '\n');

                printf("Os %d termos da PG sao:\n", n_termos_pg);
                for (int i = 0; i < n_termos_pg; i++) {
                    an_pg = a1_pg * pow(q_pg, i);
                    printf("%d ", an_pg);
                    if (i < n_termos_pg - 1) {
                        printf(" | ");
                    }
                }
                printf("\n");
                break;
            }
            case 8: {
                printf("\n--- Exercício 8: Somatorio Especifico (Eq. 2) ---\n");
                int n_soma_ex8;
                long long somatorio_ex8 = 0;

                printf("Digite o valor de N para o somatorio (N deve ser > 5): ");
                if (scanf("%d", &n_soma_ex8) != 1) {
                    printf("Entrada Invalida para N.\n");
                    while (getchar() != '\n'); break;
                }
                while (getchar() != '\n');

                if (n_soma_ex8 <= 5) {
                    printf("O valor de N deve ser maior do que 5. [cite: 10]\n");
                } else {
                    for (int i = 5; i <= n_soma_ex8; i++) {
                        somatorio_ex8 += (2*i * i + 5*i + 3);
                    }
                    printf("O resultado do somatorio de i=5 ate N=%d para a expressao (2*i^2 + 5*i + 3) e: %lld\n", n_soma_ex8, somatorio_ex8);
                }
                break;
            }
            case 9: {
                printf("\n--- Exercício 9: Serie de Fibonacci ---\n");
                int n_fibo;
                long long t1 = 1, t2 = 1, proximo_t;

                printf("Digite o numero de termos (N) da serie de Fibonacci a serem exibidos: ");
                if (scanf("%d", &n_fibo) != 1 || n_fibo <= 0) {
                    printf("Numero de termos Invalido. Deve ser um inteiro positivo.\n");
                    if (n_fibo <= 0 && getchar() == '\n') {
                        while (getchar() != '\n');
                    } else while (getchar() != '\n');
                    break;
                }
                while (getchar() != '\n');

                printf("Os %d primeiros termos da Serie de Fibonacci sao:\n", n_fibo);
                for (int i = 1; i <= n_fibo; i++) {
                    if (i == 1) {
                        printf("%lld", t1);
                    } else if (i == 2) {
                        printf("%lld", t2);
                    } else {
                        proximo_t = t1 + t2;
                        printf("%lld", proximo_t);
                        t1 = t2;
                    }
                }
            }
        }
    } while (op != 0);
}
```

```
134 case 7: {
135     printf("\n--- Exercício 7: Progressao Geometrica ---\n");
136     double a1_pg, q_pg, an_pg;
137     int n_termos_pg;
138
139     printf("Digite o valor inicial (a1) da PG: ");
140     if (scanf("%lf", &a1_pg) != 1) {
141         printf("Entrada Invalida para a1.\n");
142         while (getchar() != '\n'); break;
143     }
144     while (getchar() != '\n');
145
146     printf("Digite o valor da razao (q) da PG: ");
147     if (scanf("%lf", &q_pg) != 1) {
148         printf("Entrada Invalida para q.\n");
149         while (getchar() != '\n'); break;
150     }
151     while (getchar() != '\n');
152
153     printf("Digite o numero de termos (N) da PG: ");
154     if (scanf("%d", &n_termos_pg) != 1 || n_termos_pg <= 0) {
155         printf("Numero de termos Invalido. Deve ser um inteiro positivo.\n");
156         if (n_termos_pg <= 0 && getchar() == '\n') {
157             while (getchar() != '\n');
158         } else while (getchar() != '\n');
159         break;
160     }
161     while (getchar() != '\n');
162
163     printf("Os %d termos da PG sao:\n", n_termos_pg);
164     for (int i = 0; i < n_termos_pg; i++) {
165         an_pg = a1_pg * pow(q_pg, i);
166         printf("%d ", an_pg);
167         if (i < n_termos_pg - 1) {
168             printf(" | ");
169         }
170     }
171     printf("\n");
172     break;
173 }
174 case 8: {
175     printf("\n--- Exercício 8: Somatorio Especifico (Eq. 2) ---\n");
176     int n_soma_ex8;
177     long long somatorio_ex8 = 0;
```

```
178
179     printf("Digite o valor de N para o somatorio (N deve ser > 5): ");
180     if (scanf("%d", &n_soma_ex8) != 1) {
181         printf("Entrada Invalida para N.\n");
182         while (getchar() != '\n'); break;
183     }
184     while (getchar() != '\n');
185
186     if (n_soma_ex8 <= 5) {
187         printf("O valor de N deve ser maior do que 5. [cite: 10]\n");
188     } else {
189         for (int i = 5; i <= n_soma_ex8; i++) {
190             somatorio_ex8 += (2*i * i + 5*i + 3);
191         }
192         printf("O resultado do somatorio de i=5 ate N=%d para a expressao (2*i^2 + 5*i + 3) e: %lld\n", n_soma_ex8, somatorio_ex8);
193     }
194     break;
195 }
196 case 9: {
197     printf("\n--- Exercício 9: Serie de Fibonacci ---\n");
198     int n_fibo;
199     long long t1 = 1, t2 = 1, proximo_t;
200
201     printf("Digite o numero de termos (N) da serie de Fibonacci a serem exibidos: ");
202     if (scanf("%d", &n_fibo) != 1 || n_fibo <= 0) {
203         printf("Numero de termos Invalido. Deve ser um inteiro positivo.\n");
204         if (n_fibo <= 0 && getchar() == '\n') {
205             while (getchar() != '\n');
206         } else while (getchar() != '\n');
207         break;
208     }
209     while (getchar() != '\n');
210
211     printf("Os %d primeiros termos da Serie de Fibonacci sao:\n", n_fibo);
212     for (int i = 1; i <= n_fibo; i++) {
213         if (i == 1) {
214             printf("%lld", t1);
215         } else if (i == 2) {
216             printf("%lld", t2);
217         } else {
218             proximo_t = t1 + t2;
219             printf("%lld", proximo_t);
220             t1 = t2;
221         }
222     }
223 }
```

```

104 int main(void) {
105     int op;
106     do {
107         switch (op) {
108             case 0: {
109                 for (int i = 1; i <= n_fibo; i++) {
110                     proximo_t = t1 + t2;
111                     printf("%10d", proximo_t);
112                     t1 = t2;
113                     t2 = proximo_t;
114                 }
115                 if (i < n_fibo) {
116                     printf("\n");
117                 }
118                 printf("\n");
119                 break;
120             }
121             case 10: {
122                 printf("\n--- Exercício 10: Beecrowd 1073 - Quadrado dos Pares ---\n");
123                 int n_bee1073;
124
125                 printf("Digite um valor inteiro N (5 < N < 2000): ");
126                 if (scanf("%d", &n_bee1073) != 1) {
127                     printf("Entrada invalida para N.\n");
128                     while (getchar() != '\n'); break;
129                 }
130                 while (getchar() != '\n');
131
132                 if (n_bee1073 <= 5 || n_bee1073 >= 2000) {
133                     printf("Valor de N fora do intervalo permitido (5 < N < 2000). [cite: 10]\n");
134                 } else {
135                     printf("Quadrado de cada valor par de 1 ate %d:\n", n_bee1073);
136                     for (int i = 1; i <= n_bee1073; i++) {
137                         if (i % 2 == 0) {
138                             long long quadrado_ex10 = (long long)i * i;
139                             printf("%d*2 = %10d\n", i, quadrado_ex10);
140                         }
141                     }
142                     break;
143                 }
144             }
145             case -1: {
146                 printf("Saindo do programa.\n");
147                 break;
148             }
149             default: {
150                 printf("Opcao invalida! Tente novamente.\n");
151             }
152         }
153     } while (op != -1);
154
155     return 0;
156 }

```

```

252     }
253     case -1: {
254         printf("Saindo do programa.\n");
255         break;
256     }
257     default: {
258         printf("Opcao invalida! Tente novamente.\n");
259         break;
260     }
261 }
262
263 if (op != -1) {
264     printf("\nPressione Enter para continuar...");
265 }
266
267 } while (op != -1);
268
269 return 0;
270 }

```