Universidade da Beira Interior - Dep. Informática

Exame E. Recurso | Programação – LEI | 2025/2026

Possível resolução

1.

(a) (X >= 0) && (X <= -50 || X >= 100) ou X >= 100

(b) (N % 5 == 0) && (20 % N != 0) ou (N % 5 == 0) && (20 % N > 0)

(c) B = A % 1000 / 10; ou B = A / 10 % 100;

(d) P = 8 * (N % 2) – 5;

(e)

- ordem de cálculo: **3, 5, 4, 1, 2**
- valor: **F (falso/false)**

9 > 9 && (**6 < 6** || 2 <= 2)	aplicar operador 3
9 > 9 && (F || **2 <= 2**)	aplicar operador 5
9 > 9 && (**F || V**)	aplicar operador 4
9 > 9 && V	aplicar operador 1
F && V	aplicar operador 2
F	

(f)

- ordem de cálculo: **2, 5, 4, 1, 3**
- valor: **10.4**

20 / (**6 + 2**) * (1.2 + 13 % 9)	aplicar operador 2
20 / 8 * (1.2 + **13 % 9**)	aplicar operador 5
20 / 8 * (**1.2 + 4**)	aplicar operador 4
20 / 8 * 5.2	aplicar operador 1
2 * 5.2	aplicar operador 3
10.4	

2.

```
#include <stdio.h>
int main()
{
   int  M, SP, SI;
   float  A, B;

   A = 0;
   while(A < 100.0  ||  A > 500.0) {
      printf("Insira um numero real em [100.0, 500.0] ");
      scanf("%f", &A);
   }

   do{
      printf("Insira um numero real em ]500.0, 900.0[ ");
      scanf("%f", &B);
   }while(B <= 500.0  ||  B >= 900.0);

   if(A > B)
      M = (int) A;
   else
      M = (int) B;

   SP = 0;
   SI = 0;
   while(M > 0) {
      if(M % 2 == 0)
         SP = SP + M % 10;
      else
         SI = SI + M % 10;
      M = M / 10;
   }

   printf("A = %f, B = %f, SP = %d, SI = %d\n", A, B, SP, SI);
}
```

3.

```
int  soma (int M, int N)
{
   int  S, menor, k;
   if(M > N)
      menor = N;
   else
      menor = M;
   S = 0;
   for(k = menor; k <= M*N; k++) {
      if(5 % k == 0)
         S = S + k;
   }
   return S;
}
```

4.

```
void  divisoresMultiplos (int A[], int N, int K, int *numDiv, int *numMult)
{
   int k;
   *numDiv = 0;
   *numMult = 0;
   for(k = 0; k <= N-1; k++) {
      if(X[k] % K == 0)
         *numMult = *numMult + 1;
      if(K % A[k] == 0)
         *numDiv = *numDiv + 1;
   }
}
```

5.

a) X + 2 = 100800 + 2 x sizeof(int*) = 100800 + 2 x 8 = **100816**

b) V[0] = 40

c) W + 4 = 110180 + 4 x sizeof(int) = 110180 + 4 x 4 = **110196**

d) (*X)[0] = 60

e) V + 2 = 110172 + 2 x sizeof(int) = 110172 + 2 x 4 = **110180**

f) *V + 3 = 40 + 3 = **43**

g) **X + 2 = 60 + 2 = **62**

h) X[0] + 2 = 110180 + 2 x sizeof(int) = 110180 + 2 x 4 = **110188**

i) *(W – 3) = *(110180 – 3 x sizeof(int)) = *(110180 - 3 x 4) = *(110168) = **30**

j) *(*X – 4) = *(110180 – 4 x sizeof(int)) = *(110180 – 4 x 4) = *(110164) = **20**

6.

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
   int  P, N, k;
   float *XP, *XN, somaP, somaN, num;
   FILE *f, *g;
   f = fopen("dados.txt", "r");
   P = 0;
   N = 0;
   XP = (float*) malloc(P * sizeof(float));
   XN = (float*) malloc(N * sizeof(float));
   while(!feof(f))
   {
      fscanf(f, "%f", &num);
      if(num > 0){
         P = P + 1;
         XP = (float*) realloc(P * sizeof(float));
         XP[P-1] = num;
      }
      else
         if(num < 0){
            N = N + 1;
            XN = (float*) realloc(N * sizeof(float));
            XN[N-1] = num;
         }
   }
   fclose(f);
   somaP = 0.0;
   for(k = 0; k <= P – 1; k++)
      somaP = somaP + XP[k];
   somaN = 0.0;
   for(k = 0; k <= N – 1; k++)
      somaN = somaN + XN[k];
   g = fopen("resultados.txt", "w");
   fprintf(g, "Soma positivos = %f e soma negativos = %f\n", somaP, somaN);
   fclose(g);
}
```

7.

```
int  indiceNegativo (int A[], int N)
{
   int  num;
   // caso base/terminal
   if(N == 1)
      if(A[0] < 0)
         return A[0];
      else
         return 0;
   // caso geral: n > 1
   num = indiceMaior(A, N-1);
   if(A[N-1] < 0)
      if(num < 0)
         return num;
      else
         return A[N-1];
   else
      return num;
}
```

8.

```
int  segundoMenor (int *A, int N)
{
   int  k, menor, segMenor;
   menor = A[0];
   segMenor = A[0];
   for(k = 1; k <= N-1; k++)
   {
      if(A[k] < menor) {
         segMenor = menor;
         menor = A[k];
      }
      else {
         if(A[k] > menor)
            if(segMenor == menor)
               segMenor = A[k];
            else
               if(A[k] < segMenor)
                  segMenor = A[k];
      }
   }
   return segMenor;
}
```