

1ª Avaliação Oficial – Noturno

Professor	Disciplina	Data	NOTA
Dilermando Piva Jr.	Algoritmos e Lógica de Programação	05/out/2017	
Critérios			
Desligue seu celular! Questões rasuradas não serão consideradas. Prova SEM consulta. Não é permitido sair da sala antes de 20 minutos do início da avaliação ou durante a avaliação.			

Nome do Aluno	Visto
---------------	-------

1) Considere o seguinte trecho de um programa em Linguagem C:

```
main() {
    int a=8, b=2, c=3, d=4;
    int r = a + b;
    for( ; ; ){
        if( a-- < ++b) continue;
        d++; c--;
        if( d < a) break;
        r += d - a;
    }
    printf("R= %d \n", r);
    return 0;
}
```

Output
R= 10

Levando em consideração que todas as bibliotecas foram devidamente incluídas, o programa em questão resulta em qual saída na tela do computador depois de devidamente compilado e executado? (1 ponto)

2) Considere os seguintes comandos C. Eles são equivalentes?

Explique. (0,5 ponto)

```
char grau = 65;
char grau = 'A';
```

SIM.
para a linguagem C, uma variável caracter é o mesmo que um valor inteiro. Portanto ele pode receber ou o valor (do caracter, segundo a tabela ASCII) ou o próprio caracter, em aspas simples.

Trecho da Tabela Ascii

Decimal	Binary	Octal	Hex	ASCII	Decimal	Binary	Octal	Hex	ASCII
32	00100000	040	20	SP	64	01000000	100	40	@
33	00100001	041	21	!	65	01000001	101	41	A
34	00100010	042	22	"	66	01000010	102	42	B
35	00100011	043	23	#	67	01000011	103	43	C
36	00100100	044	24	\$	68	01000100	104	44	D
37	00100101	045	25	%	69	01000101	105	45	E

3) Calcule as seguintes expressões como C calcularia: (0,5 ponto)

- a) $8 * 9 + 2 = \underline{74}$
- c) $3 / 4 * 6 = \underline{0}$
- e) $15 \% 4 = \underline{3}$

- b) $6 * 3 / 4 = \underline{4}$
- d) $6.0 * 3 / 4 = \underline{4.5}$

Output
a)74
b)4
c)0
d)4.5
e)3

4) Suponha que **x1** e **x2** são duas variáveis do tipo **float** que você quer somá-las como inteiros e atribuir a soma a uma variável inteira. Construa um comando C para fazer isto. (0,5 ponto)

supondo que a variável que vai receber a soma seja **x**
int x;
x = (int)(x1+x2); ou **x = (int)x1 + (int)x2;**
ou
x1+=x2;
x=(int)x1;

5) O que o seguinte fragmento de código imprimiria se fosse parte de um programa válido? (1 ponto)

```
int i;  
for (i = 0; i < 5; i++)  
    cout << i; cout << "\n";
```

```
1 #include <iostream>  
2 using namespace std;  
3  
4 int main() {  
5     int i;  
6     for (i=0; i<5; i++)  
7         cout << i; cout << "\n";  
8     return 0;  
9 }  
10
```

Output

```
01234
```

6) O que o seguinte fragmento de código imprimiria se fosse parte de um programa válido? (1 ponto)

```
int j = 5;  
while ( ++j < 9)  
    cout << j++ << "\n";
```

```
1 #include <iostream>  
2 using namespace std;  
3  
4 int main() {  
5     int j=5;  
6     while (++j < 9)  
7         cout << j++ << "\n";  
8     return 0;  
9 }  
10
```

Output

```
6  
8
```

7) Construa expressões lógicas que representem as seguintes condições: (0,5 ponto - 0,1 ponto cada – todos + 0,1 ponto)

a) **peso** é maior ou igual a 55 mas menor que 70. → **(peso >= 55) && (peso < 70)**

b) **ch** é igual a **q** ou **Q** → **(ch == 'q') || (ch == 'Q')**

c) **x** é par mas não 26 → **((x%2)==0) && (x != 26)**

d) **doacao** está na faixa 1000–2000 ou **convidado** é 1. → **((doacao>=1000) && (doacao<=2000)) || (convidado == 1)**

8) Considere o seguinte fragmento de código: (1 ponto)

```
int linha = 0;  
char ch;  
while (cin.get(ch))  
{  
    if (ch == 'Q')  
        break;  
    if (ch != '\n')  
        continue;  
    linha++;  
}
```

Rescreva este código sem usar **break** ou **continue**.

```

1  #include <iostream>
2  #include <stdio.h>
3  using namespace std;
4
5  int main() {
6      char ch;
7      int linha=0;
8      cin.get(ch);
9      while ( ch != 'Q' ) {
10         if (ch == '\n')
11             linha++;
12         cin.get(ch);
13     }
14     return 0;
15 }
16

```

9) Dado o Programa C a seguir, responda as questões abaixo, sabendo-se C1, C2, C3, C4, C5 e C6 são comandos da Linguagem C, quaisquer. (vale 1 ponto – 0,25 cada)

```

#include <stdio.h>
int main()
{
    int A, B, C;
    /* comandos de atribuição, envolvendo as variáveis A, B e C, aqui entendidas como
    variáveis booleanas */
    /* ... */
    if (A){
        C1;
    } else {
        if (B) {
            if (C) {
                C2;
            } else {
                C3;
                C4;
            }
            C5;
        }
    }
    C6;
}

```

Lembrando-se que **falso = 0** e **verdade != 0**, responda:

a) Se A=verdade, B= verdade, C=falso, quais os comandos que serão executados?

C1 e C6

b) Se A=falso, B=verdade, C=verdade, quais os comandos que serão executados?

C2, C5 e C6

c) Se A=falso, B=verdade, C=falso, quais os comandos que serão executados?

C3, C4, C5 e C6

d) Quais os valores de A, B e C para que somente os comandos C3 e C4 sejam executados?

IMPOSSÍVEL! Não existe essa possibilidade, pois o C6 sempre será executado.

ESCOLHO(EMOS) FAZER O SEGUINTE PROGRAMA: _____

NOME DO ARQUIVO: _____

(Vale até 3,0 pontos)

1) Faça um programa em linguagem C para determinar a quantidade de números triangulares existentes dentro de um determinado intervalo. Um número inteiro positivo é triangular se ele é igual ao produto de três números naturais consecutivos. Exemplo $120=4 \times 5 \times 6$. Este programa recebe como parâmetro dois números inteiros positivos (1 a 65000) que estabelecem um intervalo (inclusive). Retorna a quantidade de números triangulares desse intervalo (inclusive).

```
#include <iostream>

using namespace std;

int main(){
    int limA, limB, qtd=0, i, x;

    cout << "Entre com o limite inferior: ";
    cin >> limA;
    cout << "Entre com o limite Superior: ";
    cin >> limB;
    for (x=limA; x<=limB; x++){
        for(i=1; i*(i+1)*(i+2)<=x; i++){
            if (i*(i+1)*(i+2)==x){
                qtd++;
                cout << x << endl;
            }
        }
    }
    cout << endl << "Total de numeros triangulares no intervalo: " << qtd;

    return 0;
}
```

Rode seu programa e verifique o resultado para os seguintes intervalos:

a) 1 a 500 → Total de números triangulares: 6

```
Entre com o limite inferior: 1
Entre com o limite Superior: 500
6
24
60
120
210
336
Total de numeros triangulares no intervalo: 6
Process exited after 7.608 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

b) 10451 a 21140 → Total de números triangulares: 6

```
Entre com o limite inferior: 10451
Entre com o limite Superior: 21140
10626
12144
13800
15600
17550
19656
Total de numeros triangulares no intervalo: 6
Process exited after 13.43 seconds with return value 0
Pressione qualquer tecla para continuar. . .
```

2) Faça um programa na linguagem C que escreva os números ímpares, divisíveis por 3, entre um intervalo a ser fornecido pelo usuário. Este programa recebe como parâmetro dois números inteiros positivos (1 a 65000) que estabelecem um intervalo (inclusive). Retorna em ordem crescente os números primos divisíveis por três dentro desse intervalo e a quantidade de números exibidos.

```

#include <iostream>

using namespace std;

int main(){
    int limA, limB, qtd=0, x;

    cout << "Entre com o limite inferior: ";
    cin >> limA;
    cout << "Entre com o limite Superior: ";
    cin >> limB;
    for (x=limA; x<=limB; x++){
        if ((x%2 != 0) && (x%3==0)){
            qtd++;
            cout << x << endl;
        }
    }
    cout << endl << "Total de numeros impares e div por 3 no intervalo: " << qtd;

    return 0;
}

```

Rode seu programa e verifique o resultado para os seguintes intervalos:

a) 111 a 500 → Total de números exibidos: 65

```

381
387
393
399
405
411
417
423
429
435
441
447
453
459
465
471
477
483
489
495
Total de numeros impares e div por 3 no intervalo: 65
Process exited after 7.126 seconds with return value 0
Pressione qualquer tecla para continuar. . .

```

b) 10451 a 12140 → Total de números exibidos: 281

```

12021
12027
12033
12039
12045
12051
12057
12063
12069
12075
12081
12087
12093
12099
12105
12111
12117
12123
12129
12135
Total de numeros impares e div por 3 no intervalo: 281
Process exited after 11.5 seconds with return value 0
Pressione qualquer tecla para continuar. . .

```

3) Faça um programa para mostrar a somatória de todos os números primos entre um determinado intervalo. O intervalo deve ser fornecido pelo usuário. Este programa recebe como parâmetros dois números inteiros positivos (1 a 65000) que estabelecem o intervalo (inclusive). Retorna o valor da somatória de todos os números primos neste intervalo (inclusive).

```

#include <iostream>
using namespace std;

int main(){
    int limA, limB, soma=0, i, x, div, primo;

    cout << "Entre com o limite inferior: ";
    cin >> limA;
    cout << "Entre com o limite Superior: ";
    cin >> limB;
    for (x=limA; x<=limB; x++){
        primo = 1;
        div=0;
        if (x<=1){
            primo = 0;
        } else {
            for(i=1; i<=x; i++){
                if((x%i)==0){
                    div++;
                }
            }
            if (div!=2){
                primo=0;
            }
        }
        if (primo){
            soma+=x;
            cout << endl << x;
        }
    }
    cout << endl << "Soma dos numeros primos no intervalo: " << soma;
    return 0;
}

```

Rode seu programa e verifique o resultado para os seguintes intervalos:

a) 1 a 50 = **328**

```

Entre com o limite inferior: 1
Entre com o limite Superior: 50
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
Soma dos numeros primos no intervalo: 328
Process exited after 4.645 seconds with return value 0
Pressione qualquer tecla para continuar. . .

```

b) 1045 a 2140 = **233129**

```

1997
1999
2003
2011
2017
2027
2029
2039
2053
2063
2069
2081
2083
2087
2089
2099
2111
2113
2129
2131
2137
Soma dos numeros primos no intervalo: 233129
Process exited after 13.75 seconds with return value 0
Pressione qualquer tecla para continuar. . .

```