

RESPOSTAS DA 1ª LISTA DE LINGUAGENS DE PROGRAMAÇÃO

1ª Questão

Linguagens de alto nível: São as linguagens de programação que possuem uma estrutura e palavras-chave que são mais próximas da linguagem humana. Tornando os programas mais fáceis de serem lidos e escritos.

Linguagem de baixo nível: São linguagens que se aproximam mais da forma com que o computador representa dados e instruções. Normalmente cada ordem dada nestas linguagens representa uma instrução executada pelo microcomputador. A vantagem deste tipo de linguagem é a grande velocidade de execução dos programas e o tamanho dos mesmos que são mais compactos.

Exemplo mais conhecido: Assembly.

Diferenças: Pelo fato da linguagem de alto nível se aproximar mais da forma que o programador lê e interpreta, ela se torna uma linguagem mais fácil de programar e fazer manutenção. Entretanto, a linguagem de baixo nível têm uma performance melhor.

Compilador: É o programa que transforma o código de um programa escrito em uma linguagem de alto nível para um equivalente em linguagem de máquina. Tradução é lenta, mas a execução é rápida.

2ª Questão

PERL

```
#!/usr/bin/perl
# your code goes here
print "Digite o N desejado:\n";
my $n = <>;
my $res = 1.0;
my $i = 1;
for($i=1;$i<=$n;$i++){
  if($i%2 == 1){
  } else {
```

```

}
$shr=0;
$j=0;
}
printf "%.15f", $res;

```

C

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
int main()
{
    int i,n;
    float soma=1;
    scanf("%d", &n);
    for(i=1;i<=n;i++){
        if(i%2)
            soma=soma*(float)pow(2*i-1,2*i-1)/pow(2*i,2*i);
        else
            soma=soma+(float)pow(2*i-1,2*i-1)/pow(2*i,2*i);
    }
    printf("%f\n",soma);
    return 0;
}

```

C++

```

#include <iostream>
#include <math.h>
using namespace std;

```

```

int main()
{
int i,n;
float soma=1;
cin >> n;
for(i=1;i<=n;i++){
if(i%2)
soma=soma*(float)pow(2*i-1,2*i-1)/pow(2*i,2*i);
else
soma=soma+(float)pow(2*i-1,2*i-1)/pow(2*i,2*i);}
cout << soma << endl;
return 0;
}

```

Python

```

n=int(input("entre com o valor de n: "))
soma=1
for i in range(1,(n+1)):
if(i%2==1):
soma=soma*(((2*i)-1)**((2*i)-1))/((2*i)**(2*i))
else:
soma=soma+(((2*i)-1)**((2*i)-1))/((2*i)**(2*i))
print(soma)

```

3ª Questão

C

```

#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main() {

```

```

char nomes[10][15], aux[15];
int quantidade, i, j;
printf("Quantidade total de nomes: ");
scanf("%d", &quantidade);
for(i=0; i<quantidade; i++)
scanf("%s", nomes[i]);

for(i=0; i<quantidade; i++) {

for(j=0; j<quantidade; j++) {
}
}
printf("\n-Nomes Selecionados:-\n");
for(i=0; i<2; i++)
printf("%s\n", nomes[i]);
return 0;
}

```

C++

```

#include<stdio.h>
#include<string.h>
#include <iostream>
#include <cstdlib>
using namespace std;
int main() {
char nomes[10][15], aux[15];
int quantidade, i, j;
cout << "Quantidade total de nomes: ";
cin >> quantidade;
for(i=0; i<quantidade; i++)

```

```

cin >> nomes[i];
for(i=0; i<quantidade; i++) {
for(j=0; j<quantidade; j++) {
}
}
cout << "\n--Nomes Seleccionados:--";
for(i=0; i<2; i++)
cout << "\n" << nomes[i];
return 0;
}

Perl

#!/usr/bin/perl
print 'tamanho';
$tam= <STDIN>;
for($i=0; $i<$tam;++$i)
{
$nomes=<STDIN>;
push(@nomes, $nome);
}

my @sorted = sort { lc($a) cmp lc($b) } @nomes;
print $sorted[0];
print $sorted[1];

```

5ª Questão

Imperativa: Linguagens expressam sequências de comandos que realizam transformações sobre dados.

Ex: C, Java.

Modelo Declarativo: Linguagens que não possuem os conceitos de sequências de comandos atribuição. São divididas em:

Linguagens funcionais: ênfase em valores computados por funções.

Ex:Haskell, OCaml.

Linguagens lógicas: ênfase em axiomas lógicos.

Ex: Prolog, Planner.

Marcação: É representada por um conjunto de códigos empregados a dados e a textos, com a finalidade de adicionar informações específicas sobre esse dado ou texto.

Ex: XSLI

6ª Questão

$$\langle \text{expr} \rangle \rightarrow \langle \text{expo} \rangle * \langle \text{expr} \rangle \mid \langle \text{expo} \rangle$$
$$\langle \text{expo} \rangle \rightarrow \langle \text{base} \rangle ^ \langle \text{expo} \rangle \mid \langle \text{base} \rangle$$
$$\langle \text{base} \rangle \rightarrow A \mid B \mid C$$

7ª Questão

Modelo, padrão ou estilo de programação suportado por linguagens que agrupam certas características comuns. A classificação de linguagens em paradigmas é uma consequência de decisões de projeto que tem impacto na forma segundo a qual uma aplicação real é modelada do ponto de vista computacional.

8ª Questão

O analisador léxico (scanner) é a parte do compilador responsável por ler caracteres do programa fonte e transformá-los em uma representação conveniente para o analisador sintático. O analisador léxico lê o programa fonte caractere a caractere, agrupando os caracteres lidos para formar os símbolos básicos (tokens) da linguagem.

9ª Questão

Análise semântica: Descreve o significado das expressões, das instruções e das unidades de programas. Razões para descrever a semântica: Saber precisamente o que as instruções de uma linguagem fazem e As provas de exatidão do programa recorrem a descrição formal da semântica da linguagem. Semântica operacional : Descreve o significado de um programa pela execução de suas instruções em uma máquina, seja ela real ou simulada. As mudanças no estado da máquina (memória, registradores, etc.) definem o

significado da instrução. Especificação dos efeitos de executar um programa em uma máquina (real ou simulada): Efeitos: Mudança de estados

Estados: Valores armazenados.

Problemas: Linguagem de máquina: mudanças de estados pequenos e numerosos. Armazenamento no computador real é grande e complexo. Não são usados na semântica operacional formal. Usam linguagens intermediárias e interpretadores. Semântica axiomática: Baseada em lógica (cálculo de predicados). Propósito original: Verificação formal de programas. Axiomas ou regras de inferência são definidas para cada tipo de instrução na linguagem (para permitir transformações de expressões para outras expressões). As expressões são chamadas de asserções (predicados). Baseada em Lógica Matemática. Abordagem mais abstrata. Não especifica diretamente o significado. Especifica o que pode ser provado sobre o programa.

Existe :• Pré-condição – Uma asserção que precede imediatamente uma instrução de programa e descreve as variáveis dela nesse ponto • Pós-condição – Uma asserção que segue imediatamente a uma instrução e descreve as novas restrições a essas variáveis (e possivelmente a outras) depois da execução da instrução

Semântica denotacional: Baseado na teoria de funções recursivas. O método mais abstrato de descrição da semântica. A ideia baseia-se no fato de que há maneiras rigorosas de manipular objetos matemáticos, mas não construções de linguagens de programação. O processo de construção da especificação denotacional para uma linguagem define um objeto matemático para cada entidade da linguagem– Define uma função que mapeia instâncias das entidades da linguagem em instâncias dos correspondentes objetos matemáticos.

10ª Questão

a)

$$122*y - 144 > 144$$

$$122*y > 144 + 144$$

$$122*y > 288$$

$$y > 288 / 122$$

$$y > 2,36$$

b)

$$y=5*x-5$$

$$y+5<45$$

$$y<40$$

$$5*x-5<40$$

$$5*x<45$$

$$x<9$$

c)

$$y>2$$

$$y+2>2$$

$$y>0$$

$$y=y-2$$

$$y-2>2$$

$$y>4$$

d)

para zero iteração

$$\{i=N\}$$

$$wp=(i=i+1, \{i=N\}) = \{i+1=N\}, \text{ ou } \{i=N-1\}$$

$$\{i \leq N\}$$

e)

$$wp=(sn= sn+a, sn=n*a):$$

$$n*a=sn+a$$

$$n=sn/a + a/a$$

$$n= sn/a +1$$

$$n-1=sn/a$$

$$a(n-1)=sn$$

$$\{a=sn/(n-1)\}$$