

Questão 1)

De um modo geral, linguagens de alto-nível são aquelas que se aproximam mais da linguagem humana enquanto que as linguagens de baixo-nível se aproximam mais da linguagem de máquina.

Linguagens de alto-nível apresentam maior legibilidade ao seu código, o que facilita a compreensão e interpretação, também pode ser escrita de forma a ter uma orientação maior ao programador (apresenta boa expressividade). Possui um bom nível de abstração, onde se utiliza de estruturas mais complicadas omitindo alguns detalhes. Além disso, esse tipo de linguagem, teoricamente, independe do tipo do computador. Por ser tratar de uma linguagem um pouco mais afastada da linguagem de máquina (0 e 1) o tempo de compilação e execução do programa é maior, e geralmente, ocupa mais espaço de memória para armazenar e executar um programa.

Linguagens de baixo-nível por outro lado, apresentam um alto custo de manutenção, ocupa menos memória para armazenar e executar programas, e o tempo de compilação e execução é menor.

É importante destacar, a função do compilador que age como um software básico de um computador capaz de traduzir um código fonte em código executável, e claro, apresentar mensagens de erro (verificar análise sintática por exemplo). Depois, verificar se os tipos estão corretos, não somar int com bool (se for o caso da linguagem). e por fim executar a tradução do programa fonte para objeto (em linguagem de máquina). Isto para linguagens de alto-nível. Enquanto que nas linguagens de baixo-nível o montador é quem recebe o código fonte (em assembly) e gera o código binário.

Questão 2)

```
/*Aluno: Bruno Rodrigues Caputo*/
```

```
/*Data: 23/06/2016*/
```

```
/*Nome: Exercício 2*/
```

```
/*Comando para compilação = gcc -Wall -pedantic -o Exercício Exercício2.c */
```

```
/*Comando para execução = ./Exercício */
```

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
int n; /*Número inteiro n*/
```

```
int j, i; /*contadores*/
```

```
long double numerador=1; /*Inicialização do primeiro termo do numerador*/
```

```
long double denominador=2; /*Inicialização do primeiro termo do denominador*/
```

```
long double potNum=1; /*Inicialização da Potencia do numerador*/
```

```
long double potDen=1; /*Inicialização da Potencia do denominador*/
```

```
long double resultado=1.0/4.0; /*Inicializa resultado com o primeiro retorno da serie  
uma vez que não entrará no for principal do programa*/
```

```
scanf("%d",&n);
```

```
for(i=1; i<n; i++){
```

```

    numerador=numerador + 2; /*calcular a PA do numerador*/
    for(j=0;j<numerador;j++)
        potNum=potNum*numerador;

    denominador=denominador + 2; /*calcular a PA do denominador*/
    for(j=0;j<denominador;j++)
        potDen=potDen*denominador;

    if(i%2==0) /*alternar o sinal entre "*" ou "+"*/
        resultado=resultado * (potNum/potDen);
    else
        resultado=resultado + (potNum/potDen);

    potNum=1; /*resetar o valor de potNum para ser utilizado na próxima iteração do for*/
    potDen=1; /*resetar o valor de potDen para ser utilizado na próxima iteração do for*/
}

printf("%Lf\n", resultado);

return 0;
}
Entrada:2
Saída :0,355469
Entrada:3
Saída:0,02389
Entrada:10
Saída:0,019387

```

Questão 3)

```

/*Aluno: Bruno Rodrigues Caputo*/
/*Data: 23/06/2016*/
/*Nome: Exercício3*/
/*Comando para compilação = gcc -Wall -pedantic -o Exercício Exercício3.c */
/*Comando para execução = ./Exercício */

```

```

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

```

```

    int i;

```

```

int cont=0;/*inicialização do contador que será usado para escolher qual o fator da conta*/
char c;
char vetor[100];
char PrimeiraS[100];/*Primeira String*/
char SegundaS[100];/*Segunda String*/

for(i=0;c!='\n';i++){/*Ler caracter por caracter até que Enter('\n') seja lido*/

    scanf("%c",&c);
    if(c==' ')
        i--;

    if(c==' '|| c=='\n'){
        cont++;
        if(cont>2){
            vetor[i]='\0';
            if(strcmp(vetor,PrimeiraS)<0){/*A função strcmp compara lexograficamente as duas strings*/
                if(strcmp(PrimeiraS,SegundaS)<0){/*Se for <0,significa que a string PrimeiraS é menor que
a SegundaS*/
                    strcpy(SegundaS,PrimeiraS);/*copia os elementos da string PrimeiraS para SegundaS*/
                    strcpy(PrimeiraS,vetor);
                }
            }
            else if(strcmp(vetor,SegundaS)<0)
                strcpy(SegundaS,vetor);
        }
        else{
            if(cont%2==0){/*Decide qual fator será usado no cálculo*/
                strcpy(SegundaS,vetor);
            }
            else
                strcpy(PrimeiraS,vetor);
        }

        i=-1;/*reseta o indice para a leitura*/
    }
    else{
        vetor[i]=c;

    }
}

printf("%s , %s\n",PrimeiraS,SegundaS);

```

```
return 0;
```

```
}
```

Entrada:marcus, ricardo, romulo, brauliro, jorge, marcelo, ana, fabrício

Saída:ana,brauliro

Entrada:maria,pedro,rafaela,carlos,roberto

Saída:carlos,maria

Entrada:bruna,paulo,robson,talita,maria,santos,tiago

Saída:bruna,maria

Questão 5)

Linguagens Imperativas são influenciadas pela arquitetura do computador,na execução sequencial baseada em comandos, e no armazenamento de dados alterável,conceito que vem por raiz do modo como os computadores executam os programas em linguagem de máquina.

EX:C,Pascal

Linguagens Funcionais tem como proposta um metodo básico de computação que é a aplicação de funções matemáticas à argumentos.

EX:haskell,Alice

Linguagens lógicas utilizam o uso de lógica em sua essência,com sentencias instrumentais comuns da lógica formal,de maneira geral , para determinar o caracter verdadeiro ou falso de uma sentença.

EX:Prolog,planner

Linguagens de marcação é um grupo de códigos que se formam em texto ou dados com o objetivo de adicionar informações sobre um determinado assunto.

EX:Html,Kml

Questão 6)

$\langle \text{expr} \rangle ::= \langle \text{expo} \rangle * \langle \text{expr} \rangle | \langle \text{expo} \rangle$

$\langle \text{expo} \rangle ::= \langle \text{base} \rangle ^ \langle \text{expo} \rangle | \langle \text{base} \rangle$

$\langle \text{base} \rangle ::= A | B | C$

Questão 7)

Paradigma de programação é um modelo de características que servem para categorizar um grupo de linguagens.

Questão 8)

Analisador léxico é a parte do compilador responsável por ler os caracteres do programa fonte e transforma-lô em uma representação adequada para o analisador sintático.agrupando os caracteres para formar os símbolos básicos da linguagem.

Questão 9)

Análise semântica se dá pelo entendimento do significado das expressões, das instruções e das unidades de um programa., semântica operacional se resolve a como o programa será executado e quais operações serão realizadas, enquanto, que a semântica denotacional se resume a qual o significado do programa, em que meios matemáticos ele é representado. Por fim, a semântica axiomática vê quais expressões lógicas são aceitas para um programa.

Questão 10)

Programa -A

Pré-condição:

$$x > 144$$

$$122 * y - 144 > 144$$

$$122y > 288$$

$$y > 2,36$$

Programa -B

Pré-condição:

$$x < 45$$

$$y + 5 < 45$$

$$y < 40$$

$$5 * x - 5 < 40$$

$$x < 45/5$$

$$x < 9$$

Programa -C

Pré-condição:

$$y > 2$$

$$y + 2 > 2$$

$$y > 0$$

$$y - 2 > 2$$

$$y > 4$$

Como $y > 0 \rightarrow y > 4$ então pela regra de consequência

$$y > 0$$

Programa -D

Pré-condição

$$i < N$$

Se $P = I$, então $P \rightarrow I$.

$i + 1 < N$ que é equivalente a $i < N$

O laço termina em $i = N$

Pelos passos anteriores temos que $N=0$.