Linguagem de Programação – P1 Profº Arakaki

1.

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

$$a_{i+1} = \frac{a_i x}{i}$$

```
#include <stdio.h>
#include <math.h>
int main(){
   int a1, a2 = 1, x, i = 1;
   double s, ex;
   scanf("%d", &x);
   ex = 1.00 + x;
   while (i! = x-1){
      a1 = pow(x, i);
      a2 *= i+1;
      s = (double)a1/a2;
      ex += s;
      i++;
   }
   printf("eX = %.3lf", ex);
}
```

2. Diferença entre parâmetros em tempo de execução e tempo de compilação.

Tempo de execução

É quando se define os parâmetros em tempo de execução, ou seja, quando o programa já está compilado e executando.

Tempo de compilação

É quando se define os parâmetros antes do programa ser compilado.

3.

(7 || 1) && (3 == 2) && (1 > 0) || (3 < 0)

```
  V    &&     F    &&    V    ||    F
         F          &&    V    ||    F
                     F          ||    F
                                F
```

4.

(a) Construa uma função *encaixa* que dados dois inteiros positivos *a* e *b* verifica se *b* corresponde aos últimos dígitos de *a*.

Exemplo:

a	*b*	
567890	890	=> encaixa
1243	1243	=> encaixa
2457	245	=> não encaixa
457	2457	=> não encaixa

(b) Usando a função do item anterior, faça um programa que lê dois inteiros positivos *a* e *b* e verifica se o menor deles é segmento do outro.

Exemplo:

a	*b*	
567890	678	=> b é segmento de a
1243	2212435	=> a é segmento de b
235	236	=> um não é segmento do outro

```
#include <stdio.h>
#define TRUE 1
#define FALSE 0

int encaixa(int a, int b) ;

int main() {
   int a, b, subseq, maior, menor;

   printf("Digite dois inteiros positivos: \n");
   scanf("%d %d", &a, &b);

   if (a > b) {
      maior = a;
      menor = b;
   }
   else {
      maior = b;
      menor = a;
   }

   subseq = FALSE;
   while (maior >= menor) {
      if (encaixa(maior, menor) == TRUE)
      subseq = TRUE;
   maior = maior/10;
   }
   if (subseq == TRUE)
      printf("%d eh segmento de %d\n", b, a);
   else
      printf("%d nao eh segmento de %d\n", b, a);

  return 0;
}

int encaixa(int a, int b) {

   while (b != 0 && a % 10 == b % 10) {
      a = a/10;
      b = b/10;
   }

   if (b == 0)
      return TRUE;
   else
      return FALSE;}
```

5.

```
#include<stdio.h>

int main(int argc, char *argv[]) {
   int mult = 1, c, num;

   if (argc == 1)
      printf("Falta argumentos, digite os valores apos a soma!!!\n");

   for (c = 1; c < argc; c++){
      num = atoi(argv[c]);
      mult *= num;
   }
   printf("e: %d", mult);

   return 0;
}
```

Outra resolução

```
#include <stdio.h>

int main(int argc , char * argv[]){
   int c, mult = 1;
   if (argc == 1){
      printf("Falta Argumentos, Digite os valores apos o nome do arquivo!!!");
      return 0;
   }
   for(c = 1 ;c < argc; c++){
      mult *= atoi(argv[c]);
   }
      printf("Mult: %d", mult);

      return 0;
}
```

6. Dadas duas sequências com *n* números inteiros entre 0 e 9, interpretadas como dois números inteiros de *n* algarismos, calcular a sequência de números que representa a soma dos dois inteiros. Exemplo: *n* = 8,

1ª sequência		8	2	4	3	4	2	5	1
2ª sequência	+	3	3	7	5	2	3	3	7
	1	1	6	1	8	6	5	8	8

```
#include <stdio.h>

int main(){
   int vetorA[100], i, n, inicio;
   int vetorB[100], vetorC[100];

   for(i = 0; i < 100; i++)
      vetorC[i] = 0;

   printf("Quantidade de numeros: ");
```

```
    scanf("%d", &n);
    printf("Qual eh o primeiro numero?\n");

    for(i = 1; i <= n; i++)
        scanf("%d", &vetorA[i]);

    printf("Qual eh o segundo numero?\n");

    for(i = 1; i <= n; i++)
        scanf("%d", &vetorB[i]);

    for(i = n; i > 0; i--){
        if(vetorA[i] + vetorB[i] < 10)
            vetorC[i] += vetorA[i] + vetorB[i];
        else{
            vetorC[i] += vetorA[i] + vetorB[i] - 10;
            vetorC[i-1] = 1;
        }
    }

    printf("   ");
    for(i = 1; i <= n; i++)
        printf("%d ", vetorA[i]);

    printf("\n+ ");

    for(i = 1; i <= n; i++)
        printf("%d ", vetorB[i]);

    printf("\n------------------\n");

    if(vetorC[0])
        inicio = 0;
    else
        inicio = 1;

    for(i = inicio; i <= n; i++)
        printf("%d ", vetorC[i]);

    return 0;
}
```