

Digitos verificadores

CPF

Erros de digitação

- As pessoas erram ao digitar
- Erro simples 1 -> 7
- Transposição 12 -> 21
- Transposição com pulo 132 -> 231
- Combinações
- Fonéticos 16 -> 6 (fixo?)
- Supressão (tamanho fixo!)
- Inserção (tamanho fixo!)

Digitos verificadores

- Para descobrir se houve erros de digitação adicione um ou mais digitos que são derivados dos demais
- Ao inves de XYZ use XYZW e
- $W=f(X, Y, Z)$
- CPF: são 9 dígitos, foi adicionado 1 dígito função dos 9 dígitos e mais 1 digito função dos 10, total 11 digitos.

Regra para “dezena”

- Considere o número 123456789
- Considere os produtos $9*2$, $8*3$, $7*4$, ...
- Calcule a soma destes produtos:
 $9*2+8*3+7*4+\dots+1*10=210$
- A dezena do número verificador é 0 caso o resto da divisão por 11 da soma de produtos seja 0 ou 1, caso contrário a dezena equivale a subtrair de 11 este resto. (no exemplo dezena = 0)

Regra para “unidade”

- Considere o número 1234567890
- Considere os produtos $0*2, 9*3, 8*4, \dots$
- Calcule a soma destes produtos:
 $0*2+9*3+8*4+\dots+1*11=255$
- A unidade do número verificador é 0 caso o resto da divisão por 11 da soma de produtos seja 0 ou 1, caso contrário a unidade equivale a subtrair de 11 este resto. (no exemplo unidade = 9)

```
//Programa para calculo de
//digitos verificadores de CPF
#include <stdio.h>
main(){
    printf("Forneca o numero do CPF:");
    int cpf;
    scanf("%d",&cpf);
    printf("calculo dos digitos verificadores do cpf:%d",cpf);
    int d0, d1, d2, d3, d4, d5, d6, d7, d8;
    int somaprod1, somaprod2;
    int dezena, unidade;
    int restoAux;
    d0=cpf%10;      d1=cpf/10%10;      d2=cpf/100%10;
    d3=cpf/1000%10;  d4=cpf/10000%10;
    d5=cpf/100000%10; d6=cpf/1000000%10;
    d7=cpf/10000000%10; d8=cpf/100000000%10;
```

```
//calcula da dezena
```

```
somaprod1=d0*2+d1*3+d2*4+d3*5+d4*6+  
          d5*7+d6*8+d7*9+d8*10;
```

```
restoAux=somaprod1%11;
```

```
dezena=(restoAux<2) ? (0) : (11-restoAux);
```

```
//calcula da unidade
```

```
somaprod2=dezena*2+ d0*3+d1*4+d2*5+d3*6+d4*7+  
          d5*8+d6*9+d7*10+d8*11;
```

```
restoAux=somaprod2%11;
```

```
unidade=restoAux<2 ? 0 : 11-restoAux;
```

```
//saida
```

```
printf("-%d",dezena);
```

```
printf("%d\n", unidade);
```

```
}
```

Variáveis

- Tipo: int, float, double, char, ...
(quantidade de bytes)
- Endereço
- Conteúdo
- Identificador: letras, dígitos, sublinhado,
(cifrão?)
- Ler & escrever

```

#include <stdio.h>
main(){
    printf("Forneca o numero do CPF:");
    int cpf, dezena, unidade;
    scanf("%d",&cpf);
    printf("calculo dos digitos verificadores do cpf:%d",cpf);
    dezena=(cpf%10*2+cpf/10%10*3+cpf/100%10*4+
        cpf/1000%10*5+cpf/10000%10*6+
        cpf/100000%10*7+cpf/1000000%10*8+
        cpf/10000000%10*9+cpf/100000000%10*10)%11<2
    ? 0
    : 11-(cpf%10*2+cpf/10%10*3+cpf/100%10*4+
        cpf/1000%10*5+cpf/10000%10*6+
        cpf/100000%10*7+cpf/1000000%10*8+
        cpf/10000000%10*9+cpf/100000000%10*10)%11;
}

```

```

unidade=(dezena*2+ cpf%10*3+cpf/10%10*4+
          cpf/100%10*5+cpf/1000%10*6+
          cpf/10000%10*7+cpf/100000%10*8+
          cpf/1000000%10*9+cpf/10000000%10*10+
          cpf/100000000%10*11)%11<2
? 0
: 11-(dezena*2+cpf%10*3+cpf/10%10*4+
       cpf/100%10*5+cpf/1000%10*6+
       cpf/10000%10*7+cpf/100000%10*8+
       cpf/1000000%10*9+cpf/10000000%10*10+
       cpf/100000000%10*11)%11;
printf("-%d",dezena);
printf("%d", unidade);
}

```