Strings

Simone de Oliveira Santos

Universidade Federal do Ceará

23 de outubro de 2020

Sumário

Strings

- Um texto é representado por uma cadeia de caracteres.
- Efetivamente, a linguagem C não oferece um tipo caractere.
- Os caracteres são representados internamente por códigos numéricos.
- C fornece o tipo `char`, que pode armazenar inteiros "pequenos" do tamanho de 1 byte, e pode armazenar assim 256 valores distintos.

Tabela ASCII

- A correspondência entre caracteres e códigos é feita por uma tabela de códigos.
- Em geral, usa-se a tabela ASCII, mas diferentes máquinas podem usar diferentes códigos.
- Dessa forma, recomenda-se **evitar o uso explícito dos códigos** referentes a uma tabela por questões de portabilidade.
- Deve-se usar a **constante caractere** usando aspas simples.

Função que verifica se o caractere é dígito.

```
int eDigito(char c){
    if((c >= '0') && (c <= '9')){
        return 1;
    }else{
        return 0;
    }
}
```

Função que verifica se o caractere é letra.

```
int eLetra(char c){
    if((c >= 'A') && (c <= 'z')){
        return 1;
    }else{
        return 0;
    }
}
```

Strings

- São representadas por vetores do tipo `char` terminadas obrigatoriamente pelo caractere nulo '\0'.
- Para armazenar uma String deve-se reservar uma posição adicional para o caractere de final de cadeia.
- Todas as funções que manipulam Strings recebem como parâmetro um vetor de `char`, e processam a cadeia até encontrar o caractere nulo.

Inicialização de strings

- O especificador de formato para Strings é %s.
- Uma String pode ser inicializada na declaração, usando aspas duplas.

```
int main(){
    char cidade[] = ``Crateus'';
    printf("%s", cidade);
    return 0;
}
```

Lendo Strings com scanf()

```
int main(){
    char cidade[100];
    printf("Digite o nome da cidade: ");
    scanf(" %s", cidade);
    return 0;
}
```

- Observe que não é usado o & antes da variável.

Lendo Strings com scanf()

Para receber Strings com mais de uma palavra, o código deve ser modifica para aceitar esse formato.

```
int main(){
    char cidade[100];
    printf("Digite o nome da cidade: ");
    scanf(" %[^\n]", cidade);
    return 0;
}
```

Lendo Strings com scanf()

Para limitar a quantidade de caracteres que poderão ser lidos, o código deve indicar a quantidade máxima de caracteres que serão armazenados.

```
int main(){
    char cidade[100];
    printf("Digite o nome da cidade: ");
    scanf(" %99[^\n]", cidade);
    return 0;
}
```

Percorrimento de Strings

A condição de parada é chegar no caractere nulo.

```
char texto[] = ‘‘meu texto’’;
int i;
for(i = 0; texto[i] != ’\0’; i++){
    printf("%c", texto[i]);
}
```

Função que conta a quantidade de caracteres

```
int comprimento(char *s){
    int i, cont = 0;
    for(i = 0; s[i] != '\0'; i++){
        cont++;
    }
    return cont;
}
```

Prática

1. Faça uma função que recebe um caractere como parâmetro e verifique se ele é uma vogal. A função deve retornar 1 se for vogal, ou 0 caso contrário.
2. Faça uma função que receba uma string como parâmetro e verifique se ela possui alguma vogal. Use a função anterior para construir esta. A função deve retornar 1 se houver, ou 0 caso contrário.
3. Faça uma função que receba uma string como parâmetro e retorne a quantidade de vogais ela possui.