

AULA 13: STRINGS:

Come zularem não é um tipo de recurso que pode definir polosas (ou tipos), logo a lista de tipos press opções → neste logóis.

→ Em C ~~uma string é um array de carateres que é terminado com um caractere nulo (\0);~~

→ No tombo declarar, diremos através do tipo ~~char~~ reservado ao ~~vector~~ caractere;

Ex: char str[11] = "Hello"; → o vector de 11

I	T	E	S	R	E	\0	T	H	O	L
0	1	2	3	4	5	6	7	8	9	10

o 11º elemento é o caractere nulo.

→ Empurros:

- Gets (nome);

↳ Problema: Não regua, pode passar o Tombo máxime.

↳ Solução: Não é isto de manipular a string → não é só o que foi dito é reto.

- Scansf ("%s", nome); → problema de buffer overflow

↳ Problema: Não é de usgo (e não garante da utilização)

- fgets (nome, 20, stdin); → problema de buffer overflow

↳ Problema: Pausa o Tombo comente, mas não redireciona

o terminal para a string → o resultado

é fixo e o resto →

16 • OS • 24

D	S	T	Q	A	S	S
---	---	---	---	---	---	---

↳ Solução: none [std::string("m")] != "m"
se é igual.

↳ chaves e tipo | tipo | O. resultado simples string

↳ Usar %S para visualizar de '%0.'

→ Vetor multidimensional: elementos naturais são estruturas que
contêm outros tipos de dados.

↳ Cada linha representa uma string;

↳ As colunas reúnem os caracteres de cada string;

Parâmetros / Substituição em loops de repetição:

Em caso de saída de resultado, pega o que
houve mais a saída!

-D Biblioteca String:

Permite a manipulação de strings em C++.

↳ S::strncpy("rea"); // retorna o tamanho da string

↳ strcpy(~~"destino"~~, "origem"); // copia da string

↳ strcat("destino", "origem"); // concatenar a origem no destino

↳ strcmp("str1", "str2"); // comparação entre strings
retorna 0 → iguais | > → str1 > str2 | < → str1 < str2

∅ → str1 > str2;