Lógica Aplicada à Computação

Elementos básicos da programação

Slides do Prof. Rodrigo de Barros Paes

Um desafio

Desenvolver um programa que calcule a média de 3 números

Por onde começar?

Todo programa é um conjunto de instruções

Mas qual instrução a CPU executa primeiro?

Em C:

```
1   int main()
2   {
3       return 0;
4   }
```

continuando …

Definir por onde o programa começa ✓

O que queremos fazer é **calcular a média de 3 números digitados pelo usuário**, logo precisaremos:

ler os valores que o usuário digitou;

somá-los;

dividir por 3 para obter a média.

mostrar o resultado ao usuário

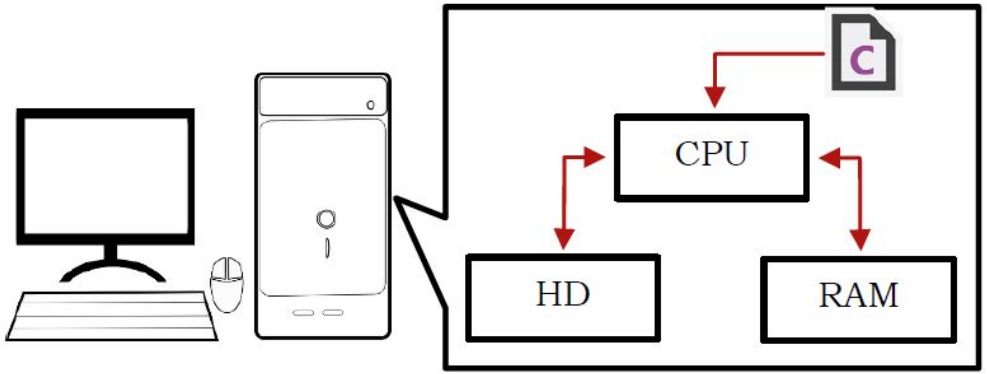
Ler os valores

Mas …

Quem vai mandar ler?

Onde vamos colocar o valor lido?

Ideias ???



Se os dados ficarão na memória, como acessá-la?

A memória

Podemos pensar nela como uma grande tabela

Acessada de forma muito eficiente pela CPU

Endereço	Valor
999999	0
1000003	0
1000007	5
1000008	‘c’
1000009	8.5
1000013	3

Essa tabela pode ser tão grande quanto for o tamanho da sua memória RAM

Para se ter uma ideia:

‘a’ = 1 byte
1 kb = 1024 bytes
1 mb = 1024 kb
1 gb = 1024 mb

Logo, um computador com 8GB de RAM pode colocar na memória:

8 * 1024 * 1024 * 1024 = 8.589.934.592 caracteres

e daí? como colocar um valor na memória?

- Opção 1:
  - Descobrir um endereço de memória livre e colocar o valor lá, “na mão”

- Opção 2:
  - utilizando variáveis

Variáveis

1. Você diz que tipo de informação você precisará armazenar para que a CPU saiba o espaço na memória ela precisará reservar pra você;

2. Depois você define um nome para poder acessar essa área na memória que a CPU reservará pra você;

3. Usando esse nome, sempre que precisar, você pode alterar o valor que está na memória.

No exemplo

O usuário digitará 3 números

Precisamos então de 3 variáveis, uma para cada número.

- Passo 1: tipo de informação
  - Números reais
- Passo 2: nomes
  - x
  - y
  - z

Em C

```
1   float x;
2   float y;
3   float z;
```

“reserve espaço na RAM para guardar um número real e chame esse espaço de x”;

“reserve espaço na RAM para guardar outro número real e chame de y”;

“reserve espaço para mais um número real e chame de z”.

Identificadores

- Os nomes das variáveis são chamados de identificadores
- Você pode utilizar qualquer sequência de uma ou mais letras, dígitos e o caractere ‘_’
- Não são permitidos espaços, pontos ou outros símbolos.
- Além disso, acentos também não são recomendados.
- Todos os identificadores devem começar com uma letra.
- Exemplos: minha_variavel, num1, x12, Y e contador.

Nossas variáveis no código e na memória

```
1   int main()
2   {
3       float x;
4       float y;
5       float z;
6       return 0;
7   }
```

Identificador	Endereço	Valor
x	1	23.34234
y	2	873.44
z	3	-234.45
	4	
	5	
	6	
	7	
	...	

Ainda sobre variáveis

Quais os tipos de variáveis válidos em C?

E porque precisamos de um tipo?

Tipo	Tipo em C	Valores válidos	Espaço necessário
inteiro	`int`	−32767 a +32767	2 bytes
inteiro (só que maiores)	`long int`	−2147483647 a +2147483647	4 bytes
caractere	`char`	qualquer caracter	1 byte
real	`float`	2^{-37} a 2^{+37}	4 bytes
real (só que mais preciso)	`double`	2^{-37} a 2^{+37}	8 bytes

Ainda sobre “ler valores”

Já temos os espaço pronto para guardar os valores, mas como ler do teclado?

scanf

Este comando permite ler um valor do teclado e armazená-lo em uma (ou mais) variável (variáveis)

No nosso caso

Variáveis x, y e z

```
1   int main()
2   {
3       float x;
4       float y;
5       float z;
6       return 0;
7   }
```

Colocando o primeiro valor digitado em x:

```
scanf("%f",&x);
```

O **scanf** é utilizado para ler valores da entrada padrão. Normalmente a entrada padrão é o teclado.

Sintaxe:

```
scanf(expressão de controle, lista de variáveis)
```

Tipo	Especificador de formato
int	%d
long int	%ld
char	%c
float	%f
double	%lf

Explorando o scanf

Suponha que iremos ler um valor int, um float e um double, poderíamos fazer:

```
int main()
{
     int a;
     float b;
     double c;

     scanf(“%d”, &a);
     scanf(“%f”,&b);
     scanf(“%lf”, &c);

     return 0;
}
```

Lendo várias variáveis em um único scanf

```
int main()
{
     int a;
     float b;
     double c;

     scanf(“%d%f%lf”, &a, &b, &c);

     return 0;
}
```

Voltando ao problema

```
1   #include<stdio.h>
2   int main()
3   {
4       float x;
5       float y;
6       float z;
7
8       scanf("%f%f%f",&x, &y, &z);
9       return 0;
10  }
```

Declarando várias variáveis de um mesmo tipo na mesma linha

```
1   #include<stdio.h>
2   int main()
3   {
4       float x, y, z;
5
6       scanf("%f%f%f",&x, &y, &z);
7       return 0;
8   }
```

Recapitulando

Definir por onde o programa começa ✓

O que queremos fazer é **calcular a média de 3 números digitados pelo usuário**, logo precisaremos:

ler os valores que o usuário digitou; ✓

somá-los;

dividir por 3 para obter a média.

mostrar o resultado ao usuário

Somar, dividir, multiplicar, subtrair: operações aritméticas

```
(x + y + z) / 3;
```

Operação	Símbolo	Sintaxe
Adição	+	`a + b`
Subtração	–	`a - b`
Multiplicação	*	`a * b`
Divisão	/	`a / b`
Resto	%	`a % b`

```
(x + y + z) / 3;
```

Valor calculado, mas como guardar esse valor?

Atribuição

Alterando os valores das variáveis

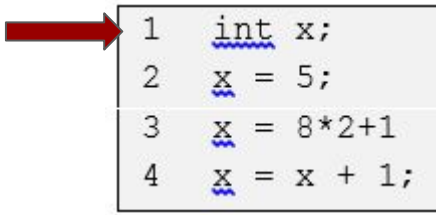
- 2 formas
  - scanf
  - atribuição
- Sintaxe:

  variável = expressão

- Exemplos

```
1   int x;
2   x = 5;
3   x = 8*2+1
4   x = x + 1;
```

Atribuição e a memória

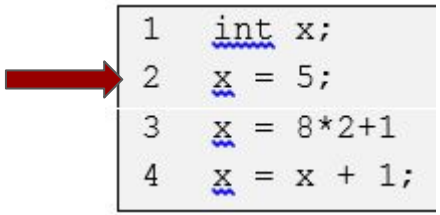


```
1   int x;
2   x = 5;
3   x = 8*2+1
4   x = x + 1;
```

lixo de memória

Identificador	Endereço	Valor
x	1324876	9873954

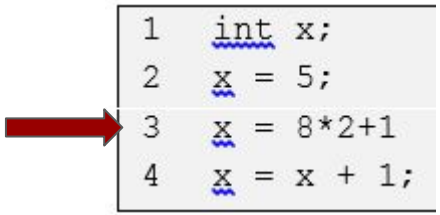
Atribuição e a memória



```
1   int x;
2   x = 5;
3   x = 8*2+1
4   x = x + 1;
```

Identificador	Endereço	Valor
x	1324876	5

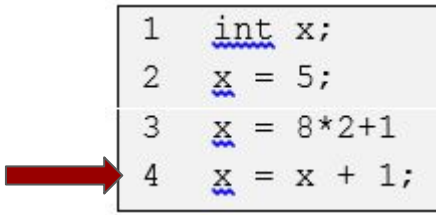
Atribuição e a memória



```
1   int x;
2   x = 5;
3   x = 8*2+1
4   x = x + 1;
```

Identificador	Endereço	Valor
x	1324876	17

Atribuição e a memória



```
1   int x;
2   x = 5;
3   x = 8*2+1
4   x = x + 1;
```

Identificador	Endereço	Valor
x	1324876	18

Voltando ao problema

1. Criar uma variável “media”
2. Atribuir o valor da expressão à ela

```
1   #include<stdio.h>
2   int main()
3   {
4       float x, y, z, media;
5       scanf("%f%f%f",&x, &y, &z);
6       media = (x + y + z) / 3;
7       return 0;
8   }
```

Recapitulando

Definir por onde o programa começa ✓

O que queremos fazer é **calcular a média de 3 números digitados pelo usuário**, logo precisaremos:

ler os valores que o usuário digitou; ✓

somá-los; ✓

dividir por 3 para obter a média. ✓

mostrar o resultado ao usuário

printf
“Imprimindo” na tela

Lembra dos dispositivos de entrada e saída?

scanf

printf

printf

Sintaxe:

```
printf( expressão de saída, parâmetros adicionais )
```

Comando	Significado
`printf("Olá pessoal");`	Imprimirá na tela: Olá pessoal
`printf("Olá pessoal\n");`	Imprimirá na tela: Olá pessoal e depois pulará uma linha por causa do caractere especial de final de linha: \n

Exemplos

```
int a, b, c;
a = 5;
b = 8;
c= a + b;
printf("A soma entre %d e %d é: %d\n", a, b, c);
```

```
int a, b;
a = 5;
b = 8;
c= a + b;
printf("A soma entre %d e %d é: %d\n", a, b, a+b);
```

```
printf("A soma entre %d e %d é: %d\n", 5, 8, 5+8);
```

Tipo	Especificador de formato
int	%d
long int	%ld
char	%c
float	%f
double	%lf

Recapitulando

Definir por onde o programa começa ✓

O que queremos fazer é **calcular a média de 3 números digitados pelo usuário**, logo precisaremos:

ler os valores que o usuário digitou; ✓

somá-los; ✓

dividir por 3 para obter a média. ✓

mostrar o resultado ao usuário ✓

A solução completa

```
1   #include<stdio.h>
2   int main()
3   {
4     float x, y, z, media;
5     scanf("%f%f%f",&x, &y, &z);
6     media = (x + y + z) / 3;
7     printf("O valor calculado foi: %f\n",media);
8     return 0;
9   }
```

O que falta?

Compilar, executar e testar!!

- Escreva esse código no editor;
- salve como calcular_media.c
- compile
  - gcc calcular_media.c -o calcular_media
- execute
  - ./calcular_media

Execute com valores diferentes, várias vezes. Deu tudo certo?

Parabéns !

Exercícios

1. Altere o programa do Código para que também imprima os números utilizados para calcular a média. Por exemplo, se os números dados foram 6 7 e 8, o seu programa deve imprimir: “O valor calculado da media entre os números 6.000000, 7.000000 e 8.000000 foi: 7.000000”.
2. Ainda no Código, experimente remover os parênteses do comando: media = (x + y + z) / 3; O que aconteceu? O valor da média ainda foi calculado corretamente? Provavelmente não né? Por que?
3. Altere o Código para calcular a média de 2 números ao invés de 3

Exercícios

4. Faça um programa que, dado 4 números, calcule o produto entre eles
5. Faça um programa que dado dois números inteiros calcule o quociente e o resto entre eles.
6. Faça um programa que leia dois números, calcule a soma entre eles, depois leia mais um número e subtraia esse número da soma calculada anteriormente. Por exemplo: se os dois primeiros números forem 5 e 8, você vai calcular a soma que dará 13, depois você lerá outro número, por exemplo, 6. E então fará 13 – 6. Você deve imprimir 7 como resposta.