



Ciência da Computação
Algoritmos e Estrutura de Dados 1

Memória, Tipos Primitivos e Ponteiros

RESPOSTAS

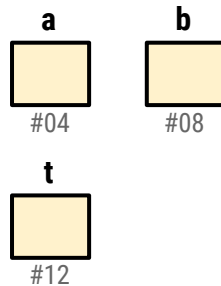
Pausa para praticar 1

Preencha as lacunas.

Código

```
int a = 20;  
int b = 30;  
int t = a;  
a = b;  
b = t;
```

Memória



Nome	Tipo	Valor	Endereço
a	int		
b	int		
t	int		

printf(“%d”, a);

printf(“%d”, b);

printf(“%d”, t);

printf(“%p”, &a);

printf(“%p”, &b);

printf(“%p”, &t);

Pausa para praticar 1

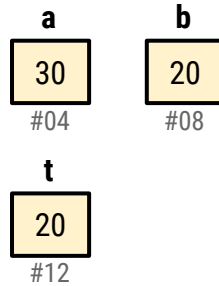
Resposta

Preencha as lacunas.

Código

```
int a = 20;  
int b = 30;  
int t = a;  
a = b;  
b = t;
```

Memória



Nome	Tipo	Valor	Endereço
a	int	30	#04
b	int	20	#08
t	int	20	#12

printf(“%d”, a);

30

printf(“%d”, b);

20

printf(“%d”, t);

20

printf(“%p”, &a);

#04

printf(“%p”, &b);

#08

printf(“%p”, &t);

#12

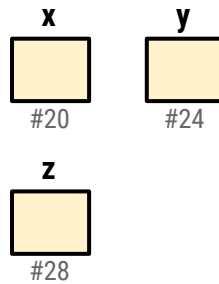
Pausa para praticar 2

Preencha as lacunas.

Código

```
int x = 10;  
int y = x++;  
int z = x;  
x *= 2;  
z = x++ + ++y;
```

Memória



Nome	Tipo	Valor	Endereço
x	int		
y	int		
z	int		

printf(“%d”, x);

printf(“%d”, y);

printf(“%d”, z);

printf(“%p”, &x);

printf(“%p”, &y);

printf(“%p”, &z);

Pausa para praticar 2

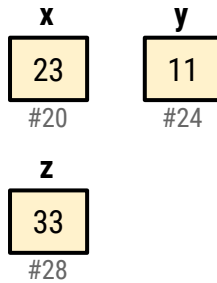
Resposta

Preencha as lacunas.

Código

```
int x = 10;  
int y = x++;  
int z = x;  
x *= 2;  
z = x++ + ++y;
```

Memória



Nome	Tipo	Valor	Endereço
x	int	23	#20
y	int	11	#24
z	int	33	#28

printf(“%d”, x);

23

printf(“%d”, y);

11

printf(“%d”, z);

33

printf(“%p”, &x);

#20

printf(“%p”, &y);

#24

printf(“%p”, &z);

#28

Exercício 1

Faça o desenho correspondente ao código abaixo utilizando a **representação simplificada da memória**.

Código

```
int i2 = 20;  
float f = 2.5;  
int *p1 = &i2;  
short int si = 30;  
short int *psi = &si;  
int **x = &p1;
```

Utilize os seguintes endereços de memória para as variáveis:

si	#12
psi	#48
p1	#32
x	#80

Desenho



Exercício 1

Resposta

Faça o desenho correspondente ao código abaixo utilizando a **representação simplificada da memória**.

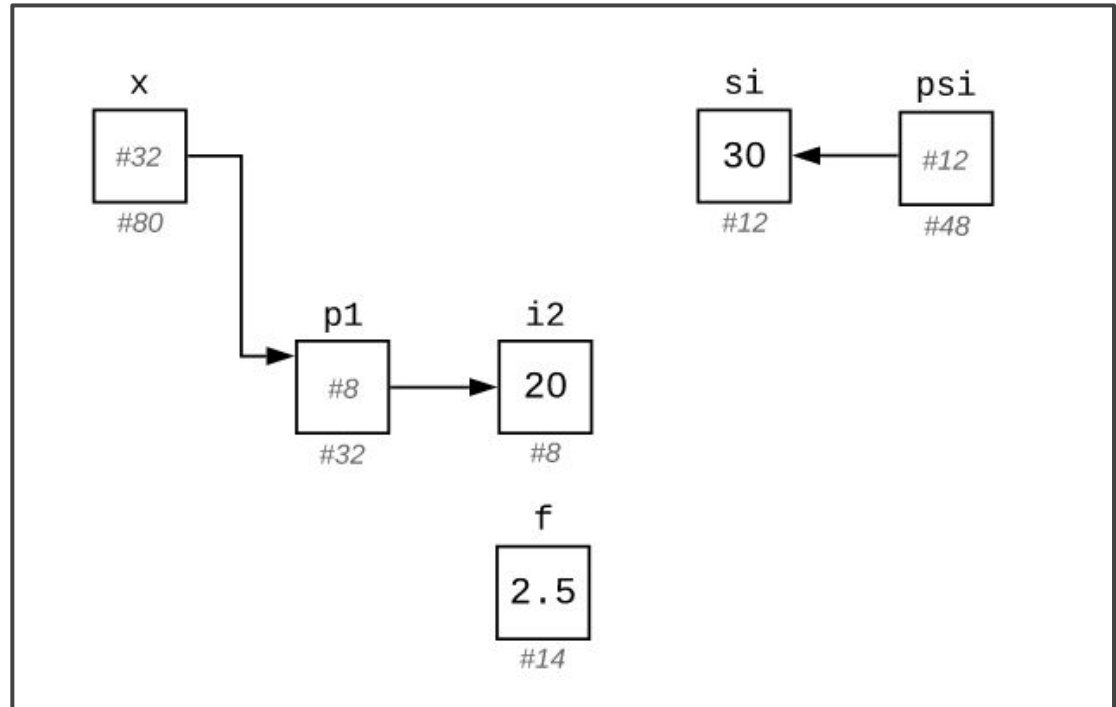
Código

```
int i2 = 20;  
float f = 2.5;  
int *p1 = &i2;  
short int si = 30;  
short int *psi = &si;  
int **x = &p1;
```

Utilize os seguintes endereços de memória para as variáveis:

si	#12
psi	#48
p1	#32
x	#80

Desenho



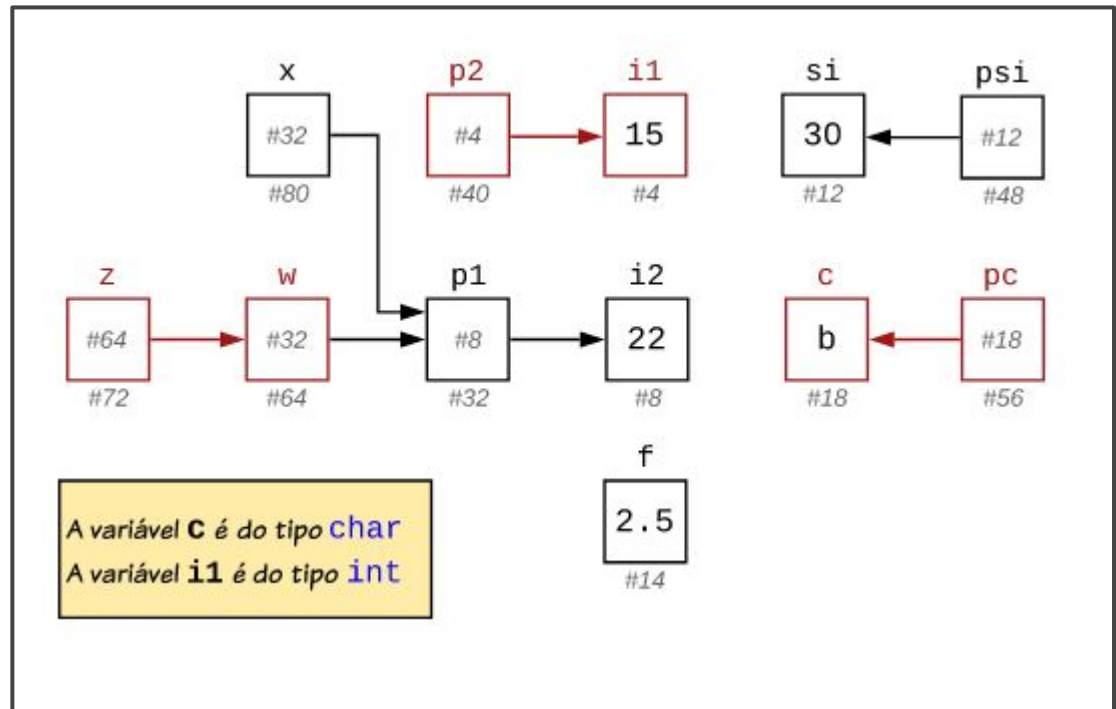
Exercício 2

A partir do resultado produzido pelo exercício anterior, **escreva o código** correspondente ao restante do desenho. (representado em vermelho)

Código

```
int i2 = 22;  
float f = 2.5;  
int *p1 = &i2;  
short int si = 30;  
short int *psi = &si;  
int **x = &p1;
```

Escreva o restante do código abaixo



Exercício 2

Resposta

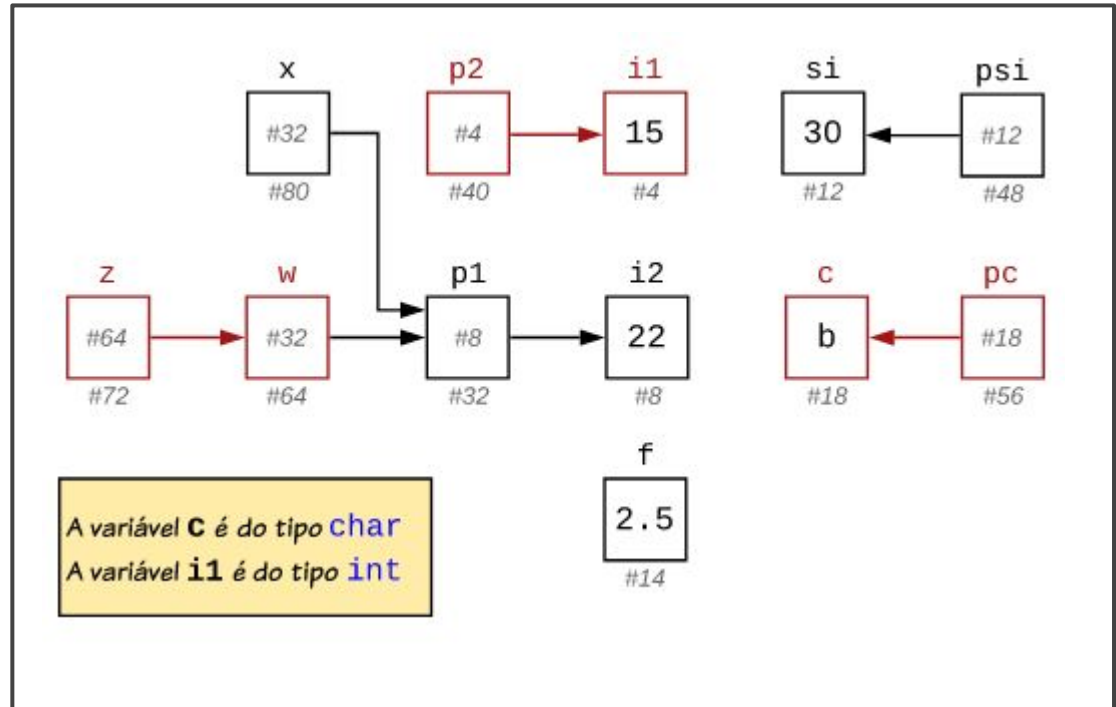
A partir do resultado produzido pelo exercício anterior, **escreva o código** correspondente ao restante do desenho. (representado em vermelho)

Código

```
int i2 = 22;  
float f = 2.5;  
int *p1 = &i2;  
short int si = 30;  
short int *psi = &si;  
int **x = &p1;
```

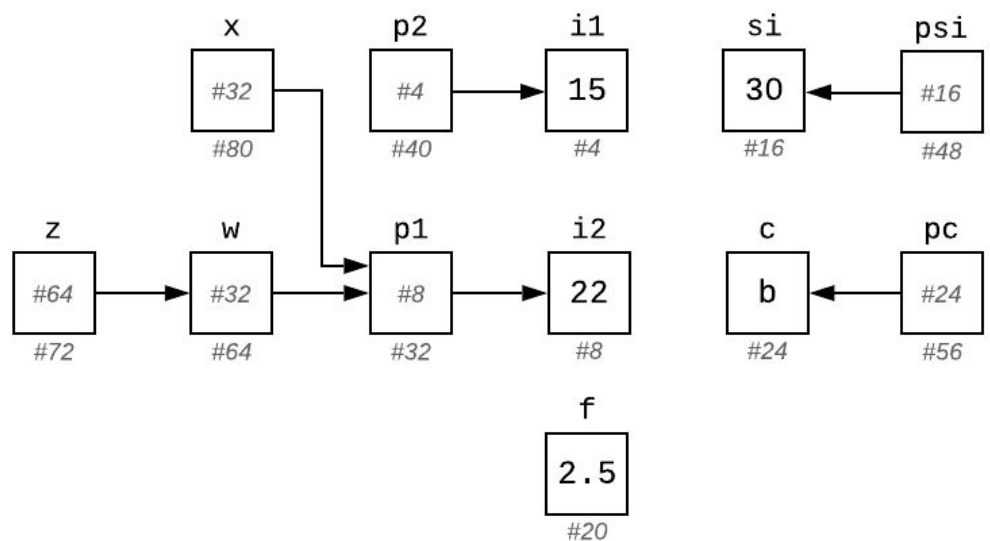
Escreva o restante do código abaixo

```
int i1 = 15;  
char c = 'b';  
int *p2 = &i1;  
char *pc = &c;  
int **w = &p1;  
int ***z = &w;
```



Exercício 3

Preencha a tabela de acordo com o código e sua representação na memória.



```
int i1 = 15;
int i2 = 22;
short int si = 30;
float f = 2.5;
char c = 'b';

int *p1 = &i2;
int *p2 = &i1;
short int *psi = &si;
char *pc = &c;
int **x = &p1;
int **w = &p1;
int ***z = &w;
```

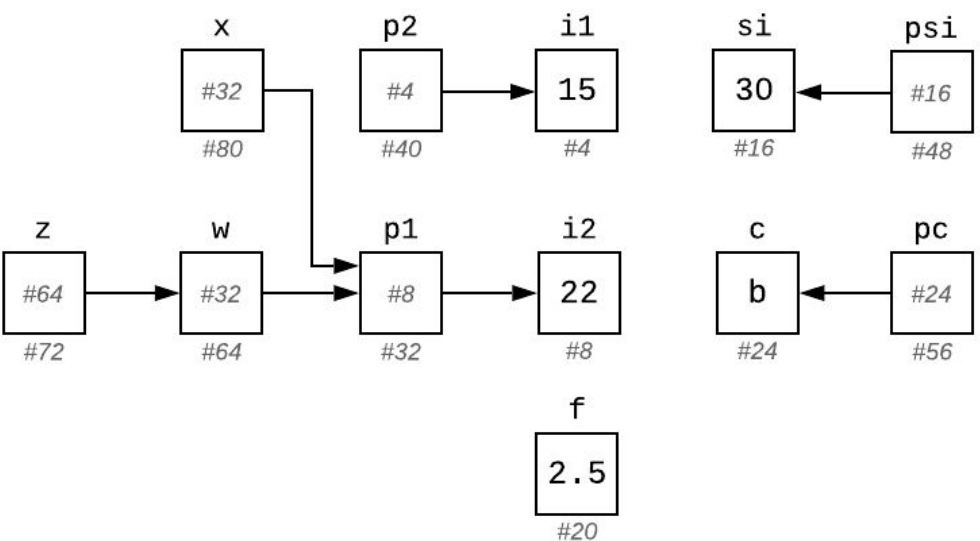
parte 1

Expressão	Valor	Tipo
i1	15	int
i2	22	int
si	30	short int
f	2.5	float
c	b	char
&i1	#4	int*
&i2		
p2		
*p2		
&p2		
z		
*z		
**z		
***z		
&z		
w		
&w		
*w		

Exercício 3

Resposta

Preencha a tabela de acordo com o código e sua representação na memória.



```
int i1 = 15;
int i2 = 22;
short int si = 30;
float f = 2.5;
char c = 'a';

int *p1 = &i2;
int *p2 = &i1;
short int *psi = &si;
char *pc = &c;
int **x = &p1;
int **w = &p1;
int ***z = &w;
```

parte 1

Expressão	Valor	Tipo
i1	15	int
i2	22	int
si	30	short int
f	2.5	float
c	b	char
&i1	#4	int*
&i2	#8	int*
p2	#4	int*
*p2	15	int
&p2	#40	int**
z	#64	int***
*z	#32	int**
**z	#8	int*
***z	22	int
&z	#72	int****
w	#32	int**
&w	#64	int***
w	#8	int