

**1)**

```
int main (void) {  
    //permite definir (ou redefinir) um tipo de dado  
    typedef struct {  
        //declara dia, mes e ano como inteiros  
        int dia, mes, ano;  
    } data; //Define o tipo de dado como sendo data  
    //mostra na tela "sizeof (data) = valor inteiro  
    printf ("sizeof (data) = %d\n",  
        //indica a quantidade de armazenamento, em bytes  
        sizeof (data));  
    //retorna o valor de saída para indicar que um programa foi encerrado com sucesso  
    return EXIT_SUCCESS;  
}
```

**2)**

```
#include <stdio.h>  
#include <stdlib.h>  
int main (void) {  
    //declara variável i como sendo inteira e recebe o valor 1234  
    int i = 1234;  
    //mostra na tela o valor de i como sendo inteiro  
    printf ("i = %d\n", i);  
    //mostra na tela o endereço armazenado de i como sendo inteiro longo  
    printf ("&i = %ld\n", (long int) &i);  
    //mostra na tela o valor em hexadecimal para o local onde o ponteiro de i  
    //está apontando  
    printf ("&i = %p\n", (void *) &i);  
    //retorna o valor de saída para indicar que um programa foi encerrado com  
    //sucesso  
    return EXIT_SUCCESS;}
```

**3)**

```
#include <stdio.h>
#include <stdlib.h>

int main (void) {
    //declara i como inteiro e ponteiro de p como inteiro
    int i; int *p;
    // i recebe 1234 e p recebe o endereço de i
    i = 1234; p = &i;
    //mostra na tela em inteiro o valor para o ponteiro de p aponta
    printf ("*p = %d\n", *p);
    //mostra na tela em um inteiro longo o endereço de i
    printf (" p = %ld\n", (long int) p);
    //mostra na tela em hexadecimal o valor do ponteiro p
    printf (" p = %p\n", (void *) p);
    //mostra em hexadecimal o endereço do ponteiro p
    printf ("&p = %p\n", (void *) &p);
    ///retorna o valor de saída para indicar que um programa foi encerrado com
    sucesso
    return EXIT_SUCCESS;
}
```

**4)**

```
int main (){
    // ponteiro de p é inteiro
    int *p ;
    // inteiro de a recebe 231
    int a = 231 ;
    // inteiro de b recebe 7680
    int b = 7680 ;
    //mostra na tela o endereço de a em hexadecimal
    printf ("&a vale %p\n", &a) ;
```

```
//mostra na tela o endereço de b em hexadecimal
printf("&b vale %p\n", &b) ;
//mostra na tela o endereço de p em hexadecimal
printf("&p vale %p\n", &p) ;
//mostra na tela o valor de p em hexadecimal
printf("p vale %p\n", p) ;
//p recebe endereço de a
p = &a ;
//mostra na tela o valor de p em hexadecimal
printf("p vale %p\n", p) ;
//mostra na tela o valor do ponteiro de p em hexadecimal
printf("**p vale %d\n", *p) ;
// p recebe o endereço de b
p = &b;
//mostra na tela o valor de p em hexadecimal
printf("p vale %p\n", p) ;
//mostra na tela o valor do ponteiro de p em hexadecimal
printf("**p vale %d\n", *p) ;
// poteiro de p recebe o valor 500
*p = 500 ;
//mostra na tela o valor de b em inteiro
printf("b vale %d\n", b) ;
//retorna 0
return 0 ;
}
```