**programas feitos em sala de aula:**

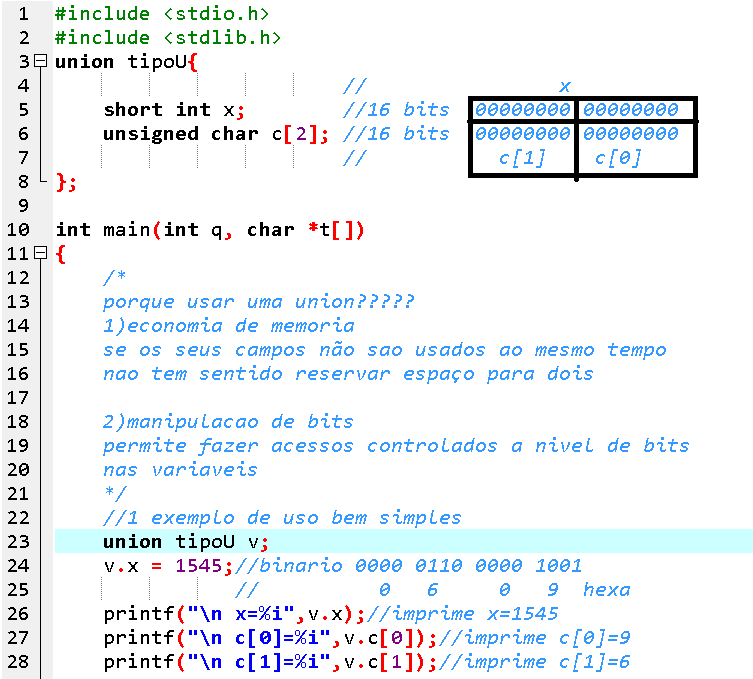
|  |  |
| --- | --- |
| **#include <stdio.h>// prog1.cpp c:>prog1 310 20 cti bauru <enter>**  **#include <stdlib.h>**  **main(int qtd, char \*txt[])**  **{**  **printf("\n qtde=%d",qtd);**  **getchar();**  **for(int j=0;j<qtd;j++)**  **{**  **printf("\n txt[%d]=%s",j,txt[j]);**  **getchar();**  **}///for**  **}**  **//vários parâmetros para dentro do programa** | **#include <stdio.h>////soma.cpp c:>soma 10 29<enter>**  **#include <stdlib.h>**  **main(int qtd, char \*txt[])**  **{**  **int valor1,valor2,valor3;**  **if(qtd!=3)**  **{**  **printf("\n erro de sintaxe: execute soma valor1 valor2<enter>");**  **return 0;**  **}**  **valor1=atoi(txt[1]);////array to int**  **valor2=atoi(txt[2]);**  **valor3=valor1+valor2;**  **printf("\n soma> %d + %d = %d",valor1,valor2,valor3);**  **}** |
| **#include<stdio.h>////copiar.cpp**  **#include<stdlib.h>**  **main(int qtd, char \*txt[])**  **{**  **char letra;**  **FILE \*arq1,\*arq2;**  **if(qtd!=3)**  **{**  **printf("\n sintaxe incorreta para copiar arquivo!!");**  **printf("\n digite:copiar arq1 arq2<enter>");**  **getchar();**  **return 0;**  **}**  **arq1=fopen(txt[1],"r");**  **if(arq1==NULL)**  **{**  **puts(" ESTE arquivo nao existe!!!");**  **puts(txt[1]);**  **return 0;**  **}** | **arq2=fopen(txt[2],"w");**  **while(1==1)**  **{**  **letra=getc(arq1);//get caracter arquivo**  **if(letra==EOF)break;///eof=end of file**  **putc(letra,arq2);**  **printf("%c",letra);**  **}**  **fclose(arq1); fclose(arq2);**  **printf("\n fimmmmmm");**  **}**  **C:> copiar arq.cpp arq2.cpp <enter>** |
| **#include <stdio.h>//hexa.cpp**  **#include <stdlib.h>**  **int main(int q, char \*t[])**  **{**  **struct hexa{**  **unsigned char h1:4; //4 bits 0000**  **unsigned char h2:4; // 4 bits 0000**  **// 1001 0001=128+16+1=145**  **// h2 h1**  **//hexa 9 1<enter>**  **// t[0] t[1] t[2]**  **}valor;**  **valor.h1=atoi(t[2]); valor.h2=atoi(t[1]);**  **printf("\n valor decimal=%i",valor);//imprime 145**  **printf("\n valor hexa=%x",valor);//imprime 91**  **printf("\n valor octal=%o",valor);//imprime 221**  **return 0;**  **}** | **#include <stdio.h>//octal.cpp**  **#include <stdlib.h>**  **int main(int q, char \*t[])**  **{**  **struct octal{**  **unsigned char o1:3; //3 bits 000**  **unsigned char o2:3; // 3 bits 000**  **unsigned char o3:2; // 2 bits 00**  **// 10 010 001**  **// o3 o2 o1**  **//octal 2 2 1<enter>**  **// t[0] t[1] t[2] t[3]**  **}valor;**  **valor.o1=atoi(t[3]); valor.o2=atoi(t[2]); valor.o3=atoi(t[1]);**  **printf("\n valor decimal=%i",valor);//imprime 145**  **printf("\n valor hexa=%x",valor);//imprime 91**  **printf("\n valor octal=%o",valor);//imprime 221**  **return 0;**  **}** |

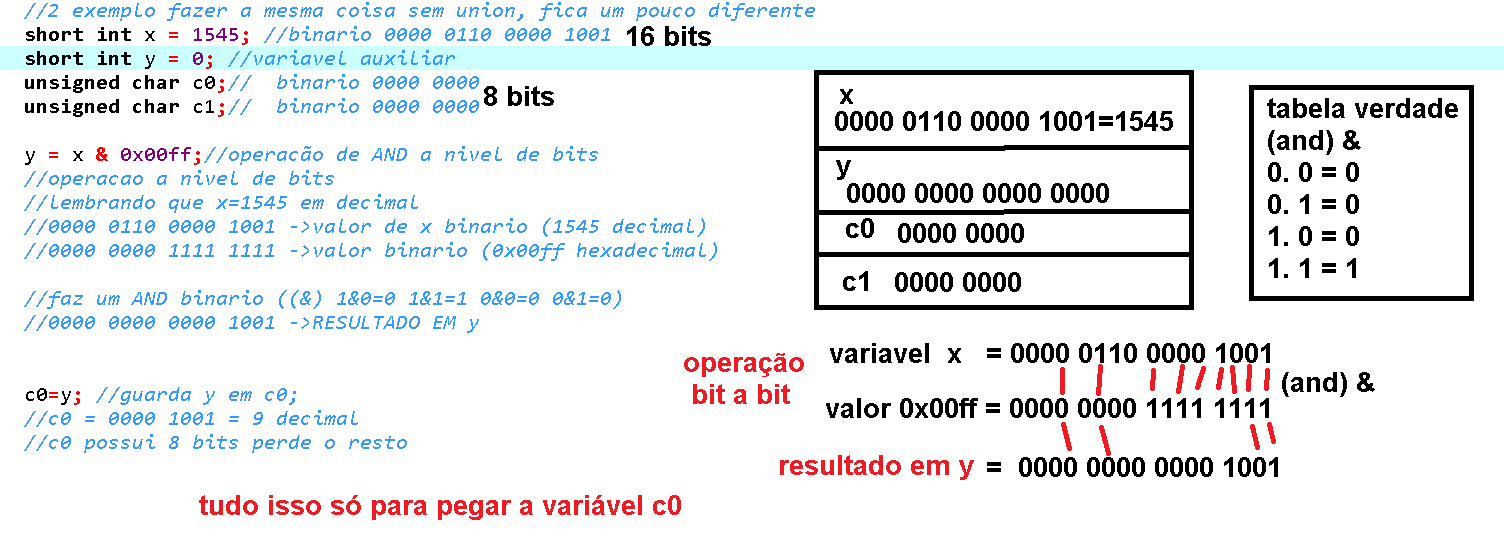
**O exemplo abaixo mostra o Uso de Union na linguagem C,**

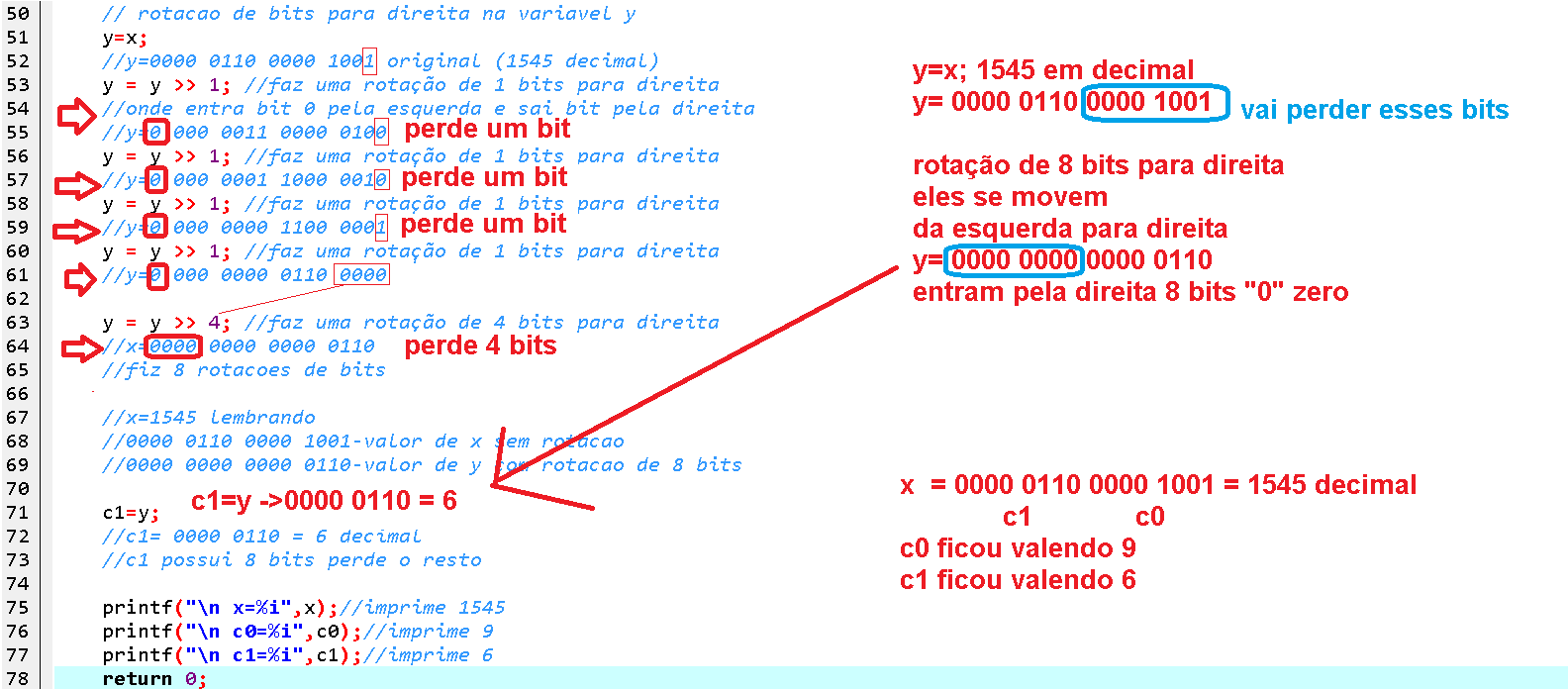
**Union facilita a manipulação de Bits em uma variável**

**para manipular bits sem union, você precisa utilizar rotação de bits (para esquerda << ou direita >>),**

**e operação and ou or a nível de bits (AND "&" ou OR "|" ), vamos ver o exemplo abaixo:**







assista os vídeos dos links abaixo:

[C] Aula 39 - Union - Parte 1

<https://www.youtube.com/watch?v=iiyjeAPeHN8>

[C] Aula 40 - Union - Parte 2

<https://www.youtube.com/watch?v=PvSTq2XHYbs>

código disponivel:

**#include <stdio.h>**

**#include <stdlib.h>**

**union tipoU{**

**// x**

**short int x; //16 bits 00000000 00000000**

**unsigned char c[2]; //16 bits 00000000 00000000**

**// c[1] c[0]**

**}v; //variavel v**

**int main(int q, char \*t[])**

**{**

**/\***

**porque usar uma union?????**

**1)economia de memoria**

**se os seus campos não sao usados ao mesmo tempo**

**nao tem sentido reservar espaço para dois**

**2)manipulacao de bits**

**permite fazer acessos controlados a nivel de bits**

**nas variaveis**

**\*/**

**//1 exemplo de uso bem simples**

**v.x = 1545;//binario 0000 0110 0000 1001**

**// 0 6 0 9 hexa**

**printf("\n x=%i",v.x);//imprime x=1545**

**printf("\n c[0]=%i",v.c[0]);//imprime c[0]=9**

**printf("\n c[1]=%i",v.c[1]);//imprime c[1]=6**

**//2 exemplo fazer a mesma coisa sem union, fica um pouco diferente**

**short int x = 1545; //binario 0000 0110 0000 1001**

**short int y = 0;**

**unsigned char c0;// binario 0000 0000**

**unsigned char c1;// binario 0000 0000**

**y = x & 0x00ff;//operacão de AND a nivel de bits**

**//operacao a nivel de bits**

**//lembrando que x=1545 em decimal**

**//0000 0110 0000 1001 ->valor de x binario (1545 decimal)**

**//0000 0000 1111 1111 ->valor binario (0x00ff hexadecimal)**

**//faz um AND binario ((&) 1&0=0 1&1=1 0&0=0 0&1=0)**

**//0000 0000 0000 1001 ->RESULTADO EM y**

**c0=y; //guarda y em c0;**

**//c0 = 0000 1001 = 9 decimal**

**//c0 possui 8 bits perde o resto**

**// rotacao de bits para direita na variavel y**

**y=x;**

**//y=0000 0110 0000 1001 original (1545 decimal)**

**y = y >> 1; //faz uma rotação de 1 bits para direita**

**//onde entra bit 0 pela esquerda e sai bit pela direita**

**//y=0 000 0011 0000 0100**

**y = y >> 1; //faz uma rotação de 1 bits para direita**

**//y=0 000 0001 1000 0010**

**y = y >> 1; //faz uma rotação de 1 bits para direita**

**//y=0 000 0000 1100 0001**

**y = y >> 1; //faz uma rotação de 1 bits para direita**

**//y=0 000 0000 0110 0000**

**y = y >> 4; //faz uma rotação de 4 bits para direita**

**//x=0000 0000 0000 0110**

**//fiz 8 rotacoes de bits**

**//x=1545 lembrando**

**//0000 0110 0000 1001-valor de x sem rotacao**

**//0000 0000 0000 0110-valor de y com rotacao de 8 bits**

**c1=y;**

**//c1= 0000 0110 = 6 decimal**

**//c1 possui 8 bits perde o resto**

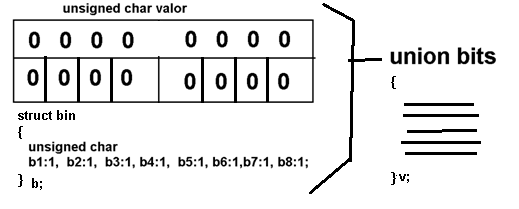
**printf("\n x=%i",x);//imprime 1545 printf("\n c0=%i",c0);//imprime 9 printf("\n c1=%i",c1);//imprime 6**

**return 0;**

**}**

**Exercício para entregar :**

**Monte um programa que utiliza UNION receba o valor pelo cmd e imprima os RESULTADOS:**



EXECUTE O PROGRAMA PELO CMD

**C:\TRABALHO> BINARIO 22<ENTER> \*obs: faça o teste para receber apenas um valor na frente do programa**

**imprimir os resultados abaixo:**

**VALOR DECIMAL =22**

**VALOR BINARIO= 0001 0110**

**HEXA=16**

**OCTAL=026**

**FIM**