

```

1  #include <stdio.h>
2  #include <math.h>
3
4  int main(){
5
6      int celula;
7      float raio, volume_nm, aresta, C;
8
9      printf("Digite o tipo da celula unitaria:\n\n1- Cubica Simples;\n2- Cubica de Corpo
Centrado (CCC);\n3- Cubica de Face Centrada (CFC);\n4- Hexagonal Simples;\n5- Hexagonal
Compacta (HC);\nTIPO: ");
10     scanf("%d", &celula);
11     printf("\nDigite o raio atomico do elemento que forma o solido em nanometros: ");
12     scanf("%f", &raio);
13
14     switch(celula){
15     case 1:
16         aresta=2*raio;
17         volume_nm=pow(aresta,3); //o VOLUME DO CUBO É SUA ARESTA AO CUBO.
18         printf("\nA celula unitaria e um cubo simples.\nVolume: %.2e A^3.\nVolume: %.2e
nm^3.", volume_nm*1000, volume_nm);
19         break;
20     case 2:
21         aresta=(4*raio)/sqrt(3); //A ARESTA É 4 VEZES O RAO DIVIDO PELA RAIZ QUADRADA DE 3.
22         volume_nm=pow(aresta,3);
23         printf("\nA celula unitaria e um Cubo de Corpo Centrado.\nVolume: %.2e A^3.\nVolume:
%.2e nm^3.", volume_nm*1000, volume_nm);
24         break;
25     case 3:
26         aresta=(4*raio)/sqrt(2); //A ARESTA É 4 VEZES O RAO DIVIDO PELA RAIZ QUADRADA DE 2.
27         volume_nm=pow(aresta,3);
28         printf("\nA celula unitaria e um Cubo de Face Centrada.\nVolume: %.2e A^3.\nVolume:
%.2e nm^3.", volume_nm*1000, volume_nm);
29         break;
30     case 4:
31         aresta=2*raio;
32         C=aresta; ////ALTURA DO HEXAGONO, O QUAL É IGUAL A ARESTA DA BASE DO HEXAGONO.
33         volume_nm=6*((pow(aresta,2)*sqrt(3))/4)*C;
34         printf("\nA celula unitaria e um Cubo de Face Centrada.\nVolume: %.2e A^3.\nVolume:
%.2e nm^3.", volume_nm*1000, volume_nm);
35         break;
36     case 5:
37         aresta=2*raio;
38         C=1.633*aresta; //A ALTURA DO HEXAGONO É 1.633 VEZES MAIOR QUE A ARESTA DA BASE.
39         volume_nm=6*((pow(aresta,2)*sqrt(3))/4)*C;
40         printf("\nA celula unitaria e um Cubo de Face Centrada.\nVolume: %.2e A^3.\nVolume:
%.2e nm^3.", volume_nm*1000, volume_nm);
41         break;
42     default:
43         printf("\nCELULA UNITARIA NAO EXISTENTE !!!");
44         break;
45     }
46
47     return 0;
48 }
49

```