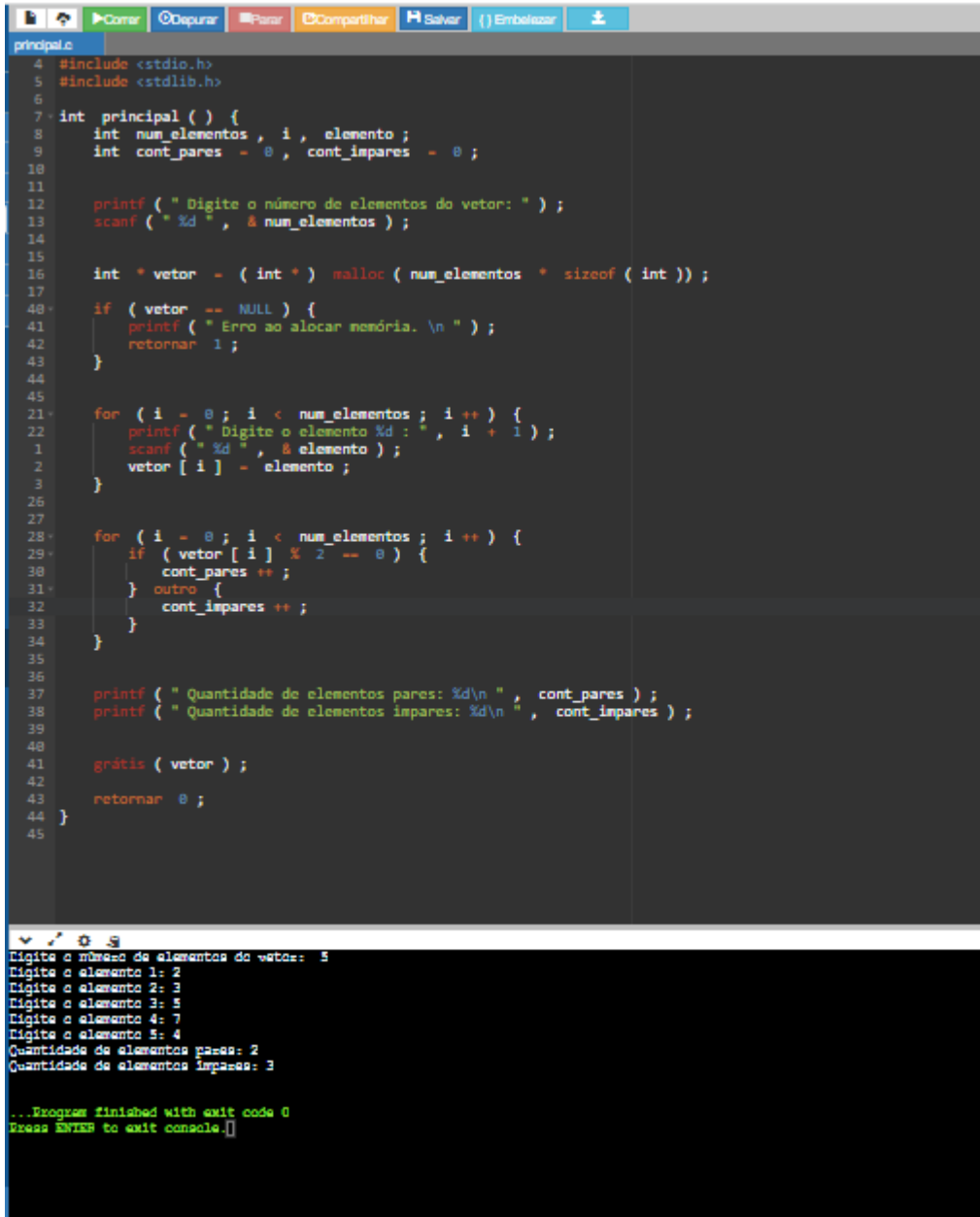


1-



The image shows a C program in a code editor with a dark theme. The program is named 'principal.c' and is located in a directory named 'principal.c'. The code defines a function 'principal()' that takes no arguments and returns an integer. It prompts the user to enter the number of elements in a vector, then enters a loop to read each element. It then counts the number of even and odd elements in the vector and prints the results. Finally, it prints the vector contents and returns 0.

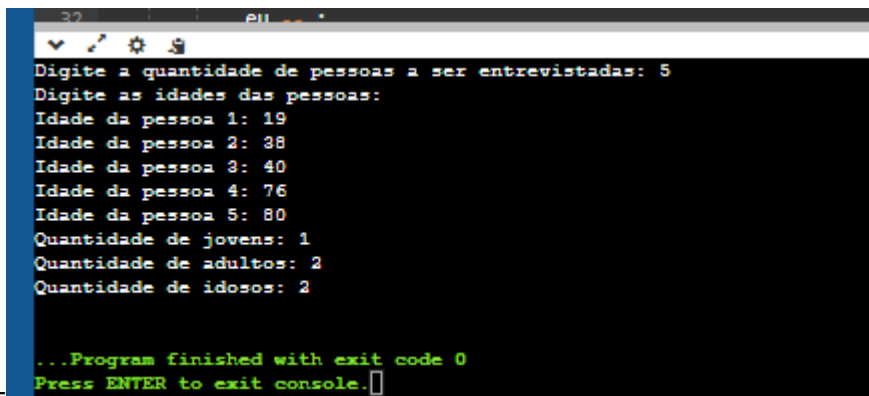
```

4 #include <stdio.h>
5 #include <stdlib.h>
6
7 int principal () {
8     int num_elementos , i , elemento ;
9     int cont_pares = 0 , cont_impares = 0 ;
10
11
12     printf ( " Digite o número de elementos do vetor: " ) ;
13     scanf ( " %d " , & num_elementos ) ;
14
15
16     int * vetor = ( int * ) malloc ( num_elementos * sizeof ( int ) ) ;
17
18     if ( vetor == NULL ) {
19         printf ( " Erro ao alocar memória. \n " ) ;
20         return 1 ;
21     }
22
23     for ( i = 0 ; i < num_elementos ; i ++ ) {
24         printf ( " Digite o elemento %d : " , i + 1 ) ;
25         scanf ( " %d " , & elemento ) ;
26         vetor [ i ] = elemento ;
27     }
28
29     for ( i = 0 ; i < num_elementos ; i ++ ) {
30         if ( vetor [ i ] % 2 == 0 ) {
31             cont_pares ++ ;
32         }
33         else {
34             cont_impares ++ ;
35         }
36     }
37
38     printf ( " Quantidade de elementos pares: %d\n " , cont_pares ) ;
39     printf ( " Quantidade de elementos impares: %d\n " , cont_impares ) ;
40
41     grátis ( vetor ) ;
42
43     return 0 ;
44 }
45

```

The console output shows the program's execution. It prompts the user to enter the number of elements (5), then enters a loop to read each element (1, 2, 3, 5, 7). It then prints the results: "Quantidade de elementos pares: 2" and "Quantidade de elementos impares: 3". Finally, it prints "Program finished with exit code 0" and "Press ENTER to exit console."

3-



The image shows a C program in a code editor with a dark theme. The program is named 'principal.c' and is located in a directory named 'principal.c'. The code defines a function 'principal()' that takes no arguments and returns an integer. It prompts the user to enter the number of people to be interviewed, then enters a loop to read each person's age. It then counts the number of young, adult, and elderly people in the group and prints the results. Finally, it prints the group contents and returns 0.

```

4 #include <stdio.h>
5 #include <stdlib.h>
6
7 int principal () {
8     int num_pessoas , i , idade ;
9     int cont_jovens = 0 , cont_adultos = 0 , cont_idosos = 0 ;
10
11
12     printf ( " Digite a quantidade de pessoas a ser entrevistadas: " ) ;
13     scanf ( " %d " , & num_pessoas ) ;
14
15
16     int * idades = ( int * ) malloc ( num_pessoas * sizeof ( int ) ) ;
17
18     if ( idades == NULL ) {
19         printf ( " Erro ao alocar memória. \n " ) ;
20         return 1 ;
21     }
22
23     for ( i = 0 ; i < num_pessoas ; i ++ ) {
24         printf ( " Digite a idade da pessoa %d : " , i + 1 ) ;
25         scanf ( " %d " , & idade ) ;
26         idades [ i ] = idade ;
27     }
28
29     for ( i = 0 ; i < num_pessoas ; i ++ ) {
30         if ( idades [ i ] < 20 ) {
31             cont_jovens ++ ;
32         }
33         else if ( idades [ i ] < 60 ) {
34             cont_adultos ++ ;
35         }
36         else {
37             cont_idosos ++ ;
38         }
39     }
40
41     printf ( " Quantidade de jovens: %d\n " , cont_jovens ) ;
42     printf ( " Quantidade de adultos: %d\n " , cont_adultos ) ;
43     printf ( " Quantidade de idosos: %d\n " , cont_idosos ) ;
44
45     grátis ( idades ) ;
46
47     return 0 ;
48 }
49

```

The console output shows the program's execution. It prompts the user to enter the number of people to be interviewed (5), then enters a loop to read each person's age (19, 38, 40, 76, 80). It then prints the results: "Quantidade de jovens: 1", "Quantidade de adultos: 2", and "Quantidade de idosos: 2". Finally, it prints "Program finished with exit code 0" and "Press ENTER to exit console."

```

principal.c
26 #include <stdio.h>
27 #include <stdlib.h>
28
29 int principal ( ) {
30     int qtd_pessoas , idade ;
31     int cont_jovens = 0 , cont_adultos = 0 , cont_idosos = 0 ;
32
33     printf ( " Digite a quantidade de pessoas a serem entrevistadas: " ) ;
34     scanf ( " %d " , & qtd_pessoas ) ;
35
36     int * idades = ( int * ) malloc ( qtd_pessoas * sizeof ( int ) ) ;
37     if ( idade == NULL ) {
38         printf ( " Erro ao alocar memória. \n " ) ;
39         retornar 1 ;
40     }
41
42     printf ( " Digite as idades das pessoas: \n " ) ;
43     for ( int i = 0 ; i < qtd_pessoas ; i ++ ) {
44         printf ( " Idade da pessoa %d : " , i + 1 ) ;
45         scanf ( " %d " , & idade ) ;
46
47         if ( idade < 18 || idade > 80 ) {
48             printf ( " Idade fora do intervalo permitido (18 a 80 anos). Tente novamente. \n " ) ;
49             eu -- ;
50             continuar ;
51         }
52
53         idades [ i ] = idade ;
54     }
55
56     for ( int i = 0 ; i < qtd_pessoas ; i ++ ) {
57         if ( idades [ i ] >= 18 && idades [ i ] < 35 ) {
58             cont_jovens ++ ;
59         } else if ( idades [ i ] >= 35 && idades [ i ] < 65 ) {
60             cont_adultos ++ ;
61         } else if ( idades [ i ] >= 65 ) {
62             cont_idosos ++ ;
63         }
64     }
65
66     printf ( " Quantidade de jovens: %d\n " , cont_jovens ) ;
67     printf ( " Quantidade de adultos: %d\n " , cont_adultos ) ;
68     printf ( " Quantidade de idosos: %d\n " , cont_idosos ) ;
69
70     |
71     grátis ( idades ) ;
72
73     retornar 0 ;
74 }
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157

```

4-

```

Digite 10 números inteiros:
Número 1: 10
Número 2: 20
Número 3: 30
Número 4: 3
Número 5: 4
Número 6: 6
Número 7: 4
Número 8: 3
Número 9: 3
Número 10: 4
Digite o valor de referência: 4
Números maiores que o valor de referência (4):
10 20 30 6
O valor de referência (4) aparece 3 vezes no vetor.

```

```

principal.c
17 #include <stdio.h>
15
16 int principal ( ) {
14     int números [ 10 ];
11     int valor_referencia ;
12     int contagem_maiores = 0 ;
13     int contagem_referencia = 0 ;
7
8
9     printf ( " Digite 10 números inteiros: \n " ) ;
10     for ( int i = 0 ; i < 10 ; i ++ ) {
26         printf ( " Número %d : ", i + 1 ) ;
2         scanf ( " %d ", & números [ i ] ) ;
3     }
4
5
6     printf ( " Digite o valor de referência: " ) ;
7     scanf ( " %d ", & valor_referencia ) ;
8
9
28     printf ( " Números maiores que o valor de referência ( %d ): \n ", valor_referencia ) ;
11     for ( int i = 0 ; i < 10 ; i ++ ) {
12         if ( números [ i ] > valor_referencia ) {
27             printf ( " %d ", números [ i ] ) ;
12             contagem_maiores ++ ;
44         }
45     }
41     if ( contagem_maiores == 0 ) {
42         printf ( " Nenhum número maior que o valor de referência. " ) ;
43     }
40     printf ( " \n " ) ;
39
38
37     for ( int i = 0 ; i < 10 ; i ++ ) {
36         if ( números [ i ] == valor_referencia ) {
35             contagem_referencia ++ ;
34         }
33     }
32
31
30     printf ( " O valor de referência ( %d ) aparece %d vezes no vetor. \n ", valor_referencia , count_referencia ) ;
29
4     retornar 0 ;
5 }
6

```

```

principal.c
41 float simulado2 ;
40 } Aluno ;
7
12 int principal ( ) {
38 int qtd_alunos ;
37
36 printf ( " Digite a quantidade de alunos: " ) ;
35 scanf ( " %d " , & qtd_alunos ) ;
11
8 Alunos alunos [ qtd_alunos ] ;
31
39 for ( int i = 0 ; i < qtd_alunos ; i ++ ) {
34 printf ( " \n Digite os dados do aluno %d : \n " , i + 1 ) ;
33
12 printf ( " Matriz: " ) ;
15 scanf ( " %d " , & alunos [ i ] .matricula ) ;
32
2 printf ( " Nome: " ) ;
26 scanf ( " %[ ^ \n] " , alunos [ i ] .nome ) ;
10
9 printf ( " Nota da AV: " ) ;
7 scanf ( " %f " , & alunos [ i ] .nota_av ) ;
6
5 printf ( " Nota do Simulado 1: " ) ;
4 scanf ( " %f " , & alunos [ i ] .simulado1 ) ;
3
8 printf ( " Nota do Simulado 2: " ) ;
9 scanf ( " %f " , & alunos [ i ] .simulado2 ) ;
28 }
11
44 printf ( " \n Dados dos alunos: \n " ) ;
30 for ( int i = 0 ; i < qtd_alunos ; i ++ ) {
29 printf ( " \n Aluno %d : \n " , i + 1 ) ;
4 printf ( " Matrícula: %d\n " , alunos [ i ] . matricula ) ;
43 printf ( " Nome: %s\n " , alunos [ i ] .nome ) ;
14 printf ( " Nota da AV: %.2f\n " , alunos [ i ] . nota_av ) ;
16 printf ( " Nota do Simulado 1: %.2f\n " , alunos [ i ] .simulado1 ) ;
17 printf ( " Nota do Simulado 2: %.2f\n " , alunos [ i ] . simulado2 ) ;
5 }
12
13 retornar 0 ;
27 }

```

5-

```

v ↗ ⚙ 🐞
Digite a quantidade de alunos: 2

Digite os dados do aluno 1:
Matrícula: 34534545
Nome: kate
Nota da AV: 50
Nota do Simulado 1: 0.8
Nota do Simulado 2: 0.9

Digite os dados do aluno 2:
Matrícula: 467474674
Nome: igor
Nota da AV: 90
Nota do Simulado 1: 1
Nota do Simulado 2: 0.5

Dados dos alunos:

Aluno 1:
Matrícula: 34534545
Nome: kate
Nota da AV: 50.00
Nota do Simulado 1: 0.80
Nota do Simulado 2: 0.90

Aluno 2:
Matrícula: 467474674
Nome: igor
Nota da AV: 90.00
Nota do Simulado 1: 1.00
Nota do Simulado 2: 0.50

...Program finished with exit code 0
Press ENTER to exit console.

```

```
principal.c
12 #include <stdio.h>
17
15 void calcularTabuada ( int num ) {
16     printf ( " Tabuada do %d : \n " , num ) ;
14     para ( int eu = 1 ; eu <= 10 ; eu ++ ) {
43         printf ( " %d x %d = %d\n " , num , i , num * i ) ;
12     }
13 }
45
27 int principal ( ) {
42     número interno ;
12     printf ( " Digite um número para calcular a tabuada: " ) ;
7     scanf ( " %d " , & número ) ;
40     calcular Tabuada ( número ) ;
41     retornar 0 ;
35 }
36
```

Digite um número para calcular a tabuada: 9
Tabuada do 9:
9 x 1 = 9
9 x 2 = 18
9 x 3 = 27
9 x 4 = 36
9 x 5 = 45
9 x 6 = 54
9 x 7 = 63
9 x 8 = 72
9 x 9 = 81
9 x 10 = 90

6-

```
principal.c
12 #include <stdio.h>
17
15 int principal ( ) {
16     float peso , altura , imc ;
14
43     printf ( " Digite seu peso em quilogramas: " ) ;
12     scanf ( " %f " , & peso ) ;
13
45     printf ( " Digite sua altura em metros: " ) ;
27     scanf ( " %f " , & altura ) ;
42
12     imc = peso / ( altura * altura ) ;
7
40     printf ( " Seu IMC é: %.2f\n " , imc ) ;
41
35     se ( imc < 18,5 ) {
36         printf ( " Classificação: Abaixo do peso \n " ) ;
37     } else if ( imc >= 18,5 && imc <= 24,9 ) {
38         printf ( " Classificação: Peso normal \n " ) ;
39     } else if ( imc >= 25 && imc <= 29,9 ) {
31         printf ( " Classificação: Sobrepeso \n " ) ;
8     } else if ( imc >= 30 && imc <= 34,9 ) {
11         printf ( " Classificação: Obesidade grau I \n " ) ;
34     } else if ( imc >= 35 && imc <= 39,9 ) {
33         printf ( " Classificação: Obesidade grau II \n " ) ;
32     } senão if ( imc >= 40 ) {
2         printf ( " Classificação: Obesidade grau III \n " ) ;
26     }
10
9     retornar 0 ;
7 }
6
```

Correr Depurar Parar Compartilhar Salvar {} Embelezar

entrada

Digite seu peso em quilogramas: 75.200
Digite sua altura em metros: 1.50
Seu IMC é: 33.42
Classificação: Obesidade grau I

7-

