



Universidade Federal de São João del Rei
Departamento de Ciência da Computação
Curso de Ciência da Computação

Roteiro 2

Adélson de Oliveira Carmo Júnior
212050019

1 Tipos Abstratos de Dados

1.1

Um Tipo Abstrato de Dados (TAD) é um modelo matemático que encapsula um determinado conjunto de dados. Na ciência da computação, sua importância é fundamental, pois permite a manipulação eficiente e dinâmica de um conjunto de dados, trazendo eficiência ao processo de programação.

1.2

Código

```
1 #ifndef __QUESTAO_1_2__
2 #define __QUESTAO_1_2__
3
4 /* struct cubo */
5 typedef struct cubo{
6     float ladoA;
7     float ladoB;
8     float ladoC;
9 }Cubo;
10
11 /* funcoes da questao */
12 float tamanhoLadoA(Cubo *cubo);
13 float tamanhoLadoB(Cubo *cubo);
14 float tamanhoLadoC(Cubo *cubo);
15 float area(Cubo *cubo);
16 float volume(Cubo *cubo);
17 #endif
18
19 questao_12/questao12.h
20
21 #include <stdio.h>
22 #include <stdlib.h>
23 #include "questao12.h"
24
25 /* Funcao que retorna o tamanho do lado
26    A do cubo */
27 float tamanhoLadoA(Cubo *cubo){
28     return cubo->ladoA;
29 }
30
31 /* Funcao que retorna o tamanho do lado
32    B do cubo */
33 float tamanhoLadoB(Cubo *cubo){
34     return cubo->ladoB;
35 }
36
37 /* Funcao que retorna o tamanho do lado
38    C do cubo */
39 float tamanhoLadoC(Cubo *cubo){
40
41     return cubo->ladoC;
42 }
43
44 /* Funcao que retorna a area da
45    superficie do cubo */
46 float area(Cubo *cubo){
47     float areaAB = cubo->ladoA *
48         cubo->ladoB;
49     float areaAC = cubo->ladoA *
50         cubo->ladoC;
51     float areaBC = cubo->ladoB *
52         cubo->ladoC;
53     return 2*areaAB + 2*areaAC +
54         2*areaBC;
55 }
56
57 /* Funcao que retorna o volume do cubo
58    */
59 float volume(Cubo *cubo){
```

```

30     return cubo->ladoA * cubo->ladoB * cubo->ladoC;
31 }
32
33 void main(){
34     Cubo *c =
        (Cubo*)malloc(sizeof(Cubo));
35     printf("Digite os lados do cubo:\n");
36     scanf("%f %f", &c->ladoA, &c->ladoB, &c->ladoC);
37     printf("O tamanho do lado A eh:
        %f\n", tamanhoLadoA(c));
38     printf("O tamanho do lado B eh:
        %f\n", tamanhoLadoB(c));
39     printf("O tamanho do lado C eh:
        %f\n", tamanhoLadoC(c));
40     printf("A area da superficie do cubo
        eh: %f\n", area(c));
41     printf("O volume do cubo eh:
        %f\n", volume(c));

```

questao_12/questao12.c

```

1 all: questao12.o
2   gcc questao12.o -o main
3
4 questao12.o: questao12.h questao12.c
5   gcc -c questao12.c

```

questao_12/Makefile

```

eadelson@adelson-junior:~/UFSJ/6_Periodo/Lab_AEDSII/roteiro3/questao12$ make
gcc questao12.o -o main
eadelson@adelson-junior:~/UFSJ/6_Periodo/Lab_AEDSII/roteiro3/questao12$ ./main
Digite os lados do cubo:
2
6
4
O tamanho do lado A eh: 2.000000
O tamanho do lado B eh: 6.000000
O tamanho do lado C eh: 4.000000
A area da superficie do cubo eh: 88.000000
O volume do cubo eh: 48.000000

```

Figura 1: Questão 1.2

1.3

Observe que nessa questão, o vetor foi ordenado a fim de facilitar uma função, a *ehigual()*. Essa ordenação foi feita por meio do quick sort, na função *qsort()*, que também usa uma função da *questao13.h*, a *compara()*.

Código

```
1 #ifndef __QUESTAO_1_3__
2 #define __QUESTAO_1_3__
3
4 /* struct do conjunto de inteiros*/
5 typedef struct conjunto_de_inteiros{
6     int *elemento;
7     int tamanho;
8 }Conjunto;
9
10 /* funcoes da questao*/
11 Conjunto* criaConjunto();
12 Conjunto* uniao(Conjunto *, Conjunto *);
13 void insereElemento(Conjunto *, int);
14 void removeElemento(Conjunto *, int);
15 Conjunto* interseccao(Conjunto *, Conjunto *);
16 Conjunto* diferenca(Conjunto *, Conjunto *);
17 int pertence(Conjunto *, int);
18 int menor(Conjunto *);
19 int maior(Conjunto *);
20 int ehigual(Conjunto *, Conjunto *);
21 int tamanho(Conjunto *);
22 int ehvazio(Conjunto *);
23 /* funcao extra*/
24 void imprimeConjunto(Conjunto *);
25 void libera(Conjunto *);
26 int compara(const void *a, const void *b);
27 void aguardaLimpa();
28
29 #endif
```

questao_13/questao13.h

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include "questao13.h"
4
5 /* Funcao que criar um conjunto vazio */
6 Conjunto* criaConjunto(){
7     Conjunto *conjunto = (Conjunto*)malloc(sizeof(Conjunto));
8     conjunto->tamanho = 0;
9     conjunto->elemento = (int*)malloc(sizeof(int)*1);
10    conjunto->elemento[0] = 999999999;
11    return conjunto;
12 }
13
14 /* Funcao que faz uniao de dois conjuntos */
15 Conjunto* uniao(Conjunto *conjuntoA, Conjunto *conjuntoB){
16     int tamanhoA = tamanho(conjuntoA);
17     int tamanhoB = tamanho(conjuntoB);
18
19     Conjunto *conjuntoC = criaConjunto();
20
21     /* Percorre o conjunto A adicionando seus elementos no conjunto C*/
```

```

22     for(int i = 0; i < tamanhoA; i++)
23         insereElemento(conjuntoC, conjuntoA->elemento[i]);
24
25     /* Percorre o conjunto B adicionando seus elementos no conjunto C */
26     for(int i=0; i < tamanhoB; i++){
27         if(!pertence(conjuntoA, conjuntoB->elemento[i]))
28             insereElemento(conjuntoC, conjuntoB->elemento[i]);
29     }
30     return conjuntoC;
31 }
32
33 /* Funcao que insere um elemento no conjunto */
34 void insereElemento(Conjunto *conjunto, int numero){
35     int tamanho_local = tamanho(conjunto);
36
37     if(tamanho_local == 0){
38         conjunto->elemento[tamanho_local] = numero;
39         conjunto->tamanho ++;
40     }
41     else{
42         if(pertence(conjunto, numero) == 1)
43             return;
44         conjunto->elemento = (int *)realloc(conjunto->elemento, (tamanho_local +
45             1) * sizeof(int));
46         conjunto->elemento[tamanho_local] = numero;
47         conjunto->tamanho ++;
48     }
49     return;
50 }
51
52 /* Funcao que remove um elemento do conjunto */
53 void removeElemento(Conjunto *conjunto, int numero){
54     int tamanho_local = tamanho(conjunto);
55     int aux;
56     /* Loop for para percorrer o conjunto */
57     for(int i=0; i < tamanho_local; i++){
58         /* Comparacao para conferir se o numero naquela posicao eh igual
59         ao buscado */
60         if(conjunto->elemento[i] == numero){
61             /* Se o elemento a ser excluido nao estiver na ultima posicao
62             troca-o de lugar com o ultimo */
63             if(i < tamanho_local-1){
64                 aux = conjunto->elemento[tamanho_local-1];
65                 conjunto->elemento[tamanho_local-1] = conjunto->elemento[i];
66                 conjunto->elemento[i] = aux;
67             }
68
69             /*diminui o tamanho do vetor e retorna*/
70             conjunto->tamanho --;
71             conjunto->elemento = (int *)realloc(conjunto->elemento, (tamanho_local
72             -1) * sizeof(int));
73             return;
74         }
75     }
76 }
77
78 /* Funcao que faz a interseccao entre dois conjuntos */
79 Conjunto* interseccao(Conjunto *conjuntoA, Conjunto *conjuntoB){
80     int tamanhoA = tamanho(conjuntoA);

```

```

79     int tamanhoB = tamanho(conjuntoB);
80
81     Conjunto *conjuntoC = criaConjunto();
82
83     /* Loops for para percorrer os conjuntos A e B*/
84     for(int i=0; i < tamanhoA; i++){
85         for(int j=0; j<tamanhoB; j++){
86             /* Comparacao para conferir se os numeros dos elementos sao iguais*/
87             if(conjuntoA->elemento[i] == conjuntoB->elemento[j])
88                 insereElemento(conjuntoC,conjuntoA->elemento[i]);
89         }
90     }
91
92     return conjuntoC;
93 }
94
95 /* Funcao que faz a diferenca de dois conjuntos */
96 Conjunto* diferenca(Conjunto *conjuntoA, Conjunto *conjuntoB){
97     int tamanhoB = tamanho(conjuntoB);
98
99     Conjunto *conjuntoC = criaConjunto();
100
101     /* Loops for para percorrer os conjuntos A e B*/
102     for(int i=0; i < tamanhoB; i++){
103         if(!pertence(conjuntoA,conjuntoB->elemento[i]))
104             insereElemento(conjuntoC,conjuntoB->elemento[i]);
105     }
106
107     return conjuntoC;
108 }
109
110 /* Funcao que testa se um numero pertence ao conjunto */
111 int pertence(Conjunto *conjunto, int numero){
112     int tamanho_local = tamanho(conjunto);
113
114     /*Loop for para percorrer o vetor */
115     for(int i=0; i < tamanho_local; i++){
116         /* Retorna 1 se o numero estiver no conjunto*/
117         if(conjunto->elemento[i] == numero)
118             return 1;
119     }
120
121     return 0;
122 }
123
124 /* Funcao que retorna o menor valor do conjunto */
125 int menor(Conjunto *conjunto){
126     int tamanho_local = tamanho(conjunto);
127     int menor = 100000000;
128
129     /*Loop for para percorrer o vetor */
130     for(int i=0; i < tamanho_local; i++){
131         /* Retorna 1 se o numero estiver no conjunto*/
132         if(conjunto->elemento[i] < menor)
133             menor = conjunto->elemento[i];
134     }
135
136     return menor;
137 }

```

```

138
139 /* Funcao que retorna o maior valor do conjunto */
140 int maior(Conjunto *conjunto){
141     int tamanho_local = tamanho(conjunto);
142     int maior = -100000000;
143
144     /*Loop for para percorrer o vetor */
145     for(int i=0; i < tamanho_local; i++){
146         /* Retorna 1 se o numero estiver no conjunto*/
147         if(conjunto->elemento[i] > maior)
148             maior = conjunto->elemento[i];
149     }
150
151     return maior;
152 }
153
154 /* Funcao que testa se dois conjuntos sao iguais */
155 int ehigual(Conjunto *conjuntoA, Conjunto *conjuntoB){
156     int tamanhoA = tamanho(conjuntoA);
157     int tamanhoB = tamanho(conjuntoB);
158
159     /* Se os tamanhos sao diferentes, entao eles nao sao iguais */
160     if(tamanhoA != tamanhoB)
161         return 0;
162
163     /* compara cada elemento para ver se sao iguais, caso nao sejam, retorna 0*/
164     for(int i = 0; i < tamanhoA; i++){
165         if(conjuntoA->elemento[i] != conjuntoB->elemento[i])
166             return 0;
167     }
168
169     /* retorna 1 se tds os elementos forem iguais*/
170     return 1;
171 }
172
173 /* Funcao que retorna o tamanho do conjunto */
174 int tamanho(Conjunto *conjunto){
175     return conjunto->tamanho;
176 }
177
178 /* Funcao que testa se o conjunto e vazio */
179 int ehvazio(Conjunto *conjunto){
180     if(tamanho(conjunto) == 0)
181         return 1;
182     else
183         return 0;
184 }
185
186 /* funcao que imprime o conjunto */
187 void imprimeConjunto(Conjunto *conjunto){
188     int tamanho_local = tamanho(conjunto);
189
190     printf("Conjunto: (");
191     for(int i = 0; i < tamanho_local; i++){
192         if(i != 0)
193             printf(",");
194         printf("%d", conjunto->elemento[i]);
195     }
196     printf(")\n");

```

```

197 }
198
199 /* Funcao que desaloca um conjunto */
200 void libera(Conjunto *conjunto){
201     free(conjunto->elemento);
202     free(conjunto);
203 }
204
205 /* Parte da funcao do quick sort */
206 int compara(const void *a, const void *b){
207     return (*(int*)a - *(int*)b);
208 }
209
210 /* Funcao para esperar resposta do usuario e depois limpar */
211 void aguardaLimpa(){
212     getchar();
213     printf("\n\nAperte qualquer tecla para continuar\n");
214     getchar();
215     system("clear");
216 }

```

questao_13/questao13.c

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include "questao13.h"
4
5 int main(){
6     int escolha_repeticao, escolha_conjunto, numero;
7     int quantidade = 0;
8
9     Conjunto *conjuntoA = criaConjunto();
10    Conjunto *conjuntoB = criaConjunto();
11    Conjunto *conjuntoC;
12
13    do{
14        printf("\nDigite o que deseja fazer: \n1- Inserir um numero no conjunto"
15              "\n2- Remover um numero no conjunto \n3- Ver o tamanho do conjunto"
16              "\n4- Ver se um numero pertence ao conjunto \n5- Ver qual o maior numero
17              do conjunto"
18              "\n6- Ver qual o menor numero do conjunto \n7- Ver a uniao entre dois
19              conjuntos"
20              "\n8- Ver a interseccao entre dois conjuntos \n9- Ver a diferenca entre
21              dois conjuntos "
22              "\n10- Ver se os conjuntos sao iguais \n11- Ver se um conjunto eh vaziao"
23              "\n12- Imprimir conjunto \n0- Sair\n");
24        scanf("%d",&escolha_repeticao);
25
26        switch (escolha_repeticao){
27
28            /* FINALIZA */
29            case 0:
30                libera(conjuntoA);
31                libera(conjuntoB);
32                break;
33
34            /* INSERCAO */
35            case 1:
36                printf("\nEscolha qual conjunto deseja fazer a insercao:\n1- Conjunto
37                      A\n2- Conjunto B\n");

```

```

34     scanf("%d",&escolha_conjunto);
35     switch (escolha_conjunto){
36         case 1:
37             printf("\nDigite o numero que deseja inserir no conjunto A: ");
38             scanf("%d", &numero);
39             insereElemento(conjuntoA,numero);
40             qsort(conjuntoA->elemento,tamanho(conjuntoA),sizeof(int),compara);
41             system("clear");
42             break;
43         case 2:
44             printf("\nDigite o numero que deseja inserir no conjunto B: ");
45             scanf("%d", &numero);
46             insereElemento(conjuntoB,numero);
47             qsort(conjuntoB->elemento,tamanho(conjuntoB),sizeof(int),compara);
48             system("clear");
49             break;
50         default:
51             printf("Valor invalido");
52             system("clear");
53             break;
54     }
55     break;
56
57     /* REMOCAO */
58     case 2:
59         printf("\nEscolha qual conjunto deseja fazer a remocao:\n1- Conjunto
60             A\n2- Conjunto B\n");
61         scanf("%d",&escolha_conjunto);
62         switch (escolha_conjunto){
63             case 1:
64                 printf("\nDigite o numero que deseja remover do conjunto A: ");
65                 scanf("%d", &numero);
66                 removeElemento(conjuntoA,numero);
67                 system("clear");
68                 break;
69             case 2:
70                 printf("\nDigite o numero que deseja remover do conjunto B: ");
71                 scanf("%d", &numero);
72                 removeElemento(conjuntoB,numero);
73                 system("clear");
74                 break;
75             default:
76                 printf("Valor invalido");
77                 system("clear");
78                 break;
79         }
80         break;
81
82     /* TAMANHO */
83     case 3:
84         printf("\nEscolha qual conjunto deseja fazer ver o tamanho:\n1-
85             Conjunto A\n2- Conjunto B\n");
86         scanf("%d",&escolha_conjunto);
87         switch (escolha_conjunto){
88             case 1:
89                 printf("\nO tamanho do conjunto A eh: %d", tamanho(conjuntoA));
90                 aguardaLimpa();
91                 break;
92             case 2:

```



```

91         printf("\n0 tamanho do conjunto B eh: %d", tamanho(conjuntoB));
92         aguardaLimpa();
93         break;
94     default:
95         printf("Valor invalido");
96         system("clear");
97         break;
98     }
99     break;
100
101     /* PERTENCE */
102     case 4:
103         printf("\nEscolha qual conjunto deseja ver se o numero pertence:\n1-
104             Conjunto A\n2- Conjunto B\n");
105         scanf("%d",&escolha_conjunto);
106         switch (escolha_conjunto){
107             case 1:
108                 printf("\nDigite o numero que deseja ver se pertence ao
109                     conjunto A: ");
110                 scanf("%d", &numero);
111                 if(pertence(conjuntoA, numero) == 1)
112                     printf("\n0 numero %d pertence ao conjunto A", numero);
113                 else
114                     printf("\n0 numero %d nao pertence ao conjunto A", numero);
115                 aguardaLimpa();
116                 break;
117             case 2:
118                 printf("\nDigite o numero que deseja ver se pertence ao
119                     conjunto B: ");
120                 scanf("%d", &numero);
121                 if(pertence(conjuntoB, numero) == 1)
122                     printf("\n0 numero %d pertence ao conjunto B", numero);
123                 else
124                     printf("\n0 numero %d nao pertence ao conjunto B", numero);
125                 aguardaLimpa();
126                 break;
127             default:
128                 printf("Valor invalido");
129                 system("clear");
130                 break;
131         }
132     }
133     break;
134
135     /* MAIOR */
136     case 5:
137         printf("\nEscolha qual conjunto deseja saber o maior elemento:\n1-
138             Conjunto A\n2- Conjunto B\n");
139         scanf("%d",&escolha_conjunto);
140         switch (escolha_conjunto){
141             case 1:
142                 printf("\n0 maior elemento do conjunto A eh: %d",
143                     maior(conjuntoA));
144                 aguardaLimpa();
145                 break;
146             case 2:
147                 printf("\n0 maior elemento do conjunto B eh: %d",
148                     maior(conjuntoB));
149                 aguardaLimpa();

```

```

144         break;
145     default:
146         printf("Valor invalido");
147         system("clear");
148         break;
149     }
150     break;
151
152     /* MENOR */
153     case 6:
154         printf("\nEscolha qual conjunto deseja saber o menor elemento:\n1-
155             Conjunto A\n2- Conjunto B\n");
156         scanf("%d",&escolha_conjunto);
157         switch (escolha_conjunto){
158             case 1:
159                 printf("\n0 menor elemento do conjunto A eh: %d",
160                     menor(conjuntoA));
161                 aguardaLimpa();
162                 break;
163             case 2:
164                 printf("\n0 menor elemento do conjunto B eh: %d",
165                     menor(conjuntoB));
166                 aguardaLimpa();
167                 break;
168             default:
169                 printf("Valor invalido");
170                 system("clear");
171                 break;
172         }
173         break;
174
175     /* UNIAO */
176     case 7:
177         conjuntoC = uniao(conjuntoA,conjuntoB);
178         imprimeConjunto(conjuntoC);
179         aguardaLimpa();
180         libera(conjuntoC);
181         break;
182
183     /* INTERSECCAO */
184     case 8:
185         conjuntoC = interseccao(conjuntoA,conjuntoB);
186         imprimeConjunto(conjuntoC);
187         aguardaLimpa();
188         libera(conjuntoC);
189         break;
190
191     /* DIFERENCA */
192     case 9:
193         conjuntoC = diferenca(conjuntoA,conjuntoB);
194         imprimeConjunto(conjuntoC);
195         aguardaLimpa();
196         libera(conjuntoC);
197         break;
198
199     /* EH IGUAL */
200     case 10:
201         if(ehigual(conjuntoA, conjuntoB) == 1)
202             printf("\n0s conjuntos sao iguais");

```

```

200         else
201             printf("\nOs conjuntos nao sao iguais");
202         aguardaLimpa();
203         break;
204
205     /* EH IGUAL*/
206     case 11:
207         printf("\nEscolha qual conjunto deseja saber se eh vazio:\n1- Conjunto
208             A\n2- Conjunto B\n");
209         scanf("%d",&escolha_conjunto);
210         switch (escolha_conjunto){
211             case 1:
212                 if(ehvazio(conjuntoA) == 1)
213                     printf("\n0 conjunto A eh vazio");
214                 else
215                     printf("\n0 conjunto A nao eh vazio");
216                 aguardaLimpa();
217                 break;
218             case 2:
219                 if(ehvazio(conjuntoB) == 1)
220                     printf("\n0 conjunto B eh vazio");
221                 else
222                     printf("\n0 conjunto B nao eh vazio");
223                 aguardaLimpa();
224                 break;
225             default:
226                 printf("Valor invalido");
227                 system("clear");
228                 break;
229         }
230         break;
231
232     /* TAMANHO*/
233     case 12:
234         printf("\nEscolha qual conjunto deseja imprimir:\n1- Conjunto A\n2-
235             Conjunto B\n");
236         scanf("%d",&escolha_conjunto);
237         switch (escolha_conjunto){
238             case 1:
239                 imprimeConjunto(conjuntoA);
240                 aguardaLimpa();
241                 break;
242             case 2:
243                 imprimeConjunto(conjuntoB);
244                 aguardaLimpa();
245                 break;
246             default:
247                 printf("Valor invalido");
248                 system("clear");
249                 break;
250         }
251         break;
252     default:
253         printf("Valor invalido");
254         system("clear");
255         break;
256 }

```

```

257     }while(escolha_repeticao != 0);
258     return 1;
259 }

```

questao_13/main.c

```

1 all: questao13.o
2   gcc questao13.o main.c -o main
3
4 questao13.o: questao13.h questao13.c
5   gcc -c questao13.c

```

questao_13/Makefile

```

adelson@adelson-junior:~/UFSJ/6_Periodo/Lab_AEDSII/roteiro3/questao13$ make
gcc questao13.o main.c -o main
adelson@adelson-junior:~/UFSJ/6_Periodo/Lab_AEDSII/roteiro3/questao13$ ./main

Digite o que deseja fazer:
1- Inserir um numero no conjunto
2- Remover um numero no conjunto
3- Ver o tamanho do conjunto
4- Ver se um numero pertence ao conjunto
5- Ver qual o maior numero do conjunto
6- Ver qual o menor numero do conjunto
7- Ver a uniao entre dois conjuntos
8- Ver a interseccao entre dois conjuntos
9- Ver a diferenca entre dois conjuntos
10- Ver se os conjuntos sao iguais
11- Ver se um conjunto eh vazio
12- Imprimir conjunto
0- Sair
1

Escolha qual conjunto deseja fazer a insercao:
1- Conjunto A
2- Conjunto B
1

Digite o numero que deseja inserir no conjunto A: 5

```

Figura 2: Questão 1.2 - Uso da função *insereElemento*

```

Digite o que deseja fazer:
1- Inserir um numero no conjunto
2- Remover um numero no conjunto
3- Ver o tamanho do conjunto
4- Ver se um numero pertence ao conjunto
5- Ver qual o maior numero do conjunto
6- Ver qual o menor numero do conjunto
7- Ver a uniao entre dois conjuntos
8- Ver a interseccao entre dois conjuntos
9- Ver a diferenca entre dois conjuntos
10- Ver se os conjuntos sao iguais
11- Ver se um conjunto eh vazio
12- Imprimir conjunto
0- Sair
1

Escolha qual conjunto deseja fazer a insercao:
1- Conjunto A
2- Conjunto B
1

Digite o numero que deseja inserir no conjunto A: 6

```

Figura 3: Questão 1.2 - Uso da função *insereElemento*

```

Digite o que deseja fazer:
1- Inserir um numero no conjunto
2- Remover um numero no conjunto
3- Ver o tamanho do conjunto
4- Ver se um numero pertence ao conjunto
5- Ver qual o maior numero do conjunto
6- Ver qual o menor numero do conjunto
7- Ver a uniao entre dois conjuntos
8- Ver a interseccao entre dois conjuntos
9- Ver a diferenca entre dois conjuntos
10- Ver se os conjuntos sao iguais
11- Ver se um conjunto eh vazio
12- Imprimir conjunto
0- Sair
1

Escolha qual conjunto deseja fazer a insercao:
1- Conjunto A
2- Conjunto B
1

Digite o numero que deseja inserir no conjunto A: 7

```

Figura 4: Questão 1.2 - Uso da função *insereElemento*

```

Digite o que deseja fazer:
1- Inserir um numero no conjunto
2- Remover um numero no conjunto
3- Ver o tamanho do conjunto
4- Ver se um numero pertence ao conjunto
5- Ver qual o maior numero do conjunto
6- Ver qual o menor numero do conjunto
7- Ver a uniao entre dois conjuntos
8- Ver a interseccao entre dois conjuntos
9- Ver a diferenca entre dois conjuntos
10- Ver se os conjuntos sao iguais
11- Ver se um conjunto eh vazio
12- Imprimir conjunto
0- Sair
12

Escolha qual conjunto deseja imprimir:
1- Conjunto A
2- Conjunto B
1
Conjunto: (5,6,7)

Aperte qualquer tecla para continuar

```

Figura 5: Questão 1.2 - Conjunto A impresso na tela

```
Digite o que deseja fazer:
1- Inserir um numero no conjunto
2- Remover um numero no conjunto
3- Ver o tamanho do conjunto
4- Ver se um numero pertence ao conjunto
5- Ver qual o maior numero do conjunto
6- Ver qual o menor numero do conjunto
7- Ver a uniao entre dois conjuntos
8- Ver a interseccao entre dois conjuntos
9- Ver a diferenca entre dois conjuntos
10- Ver se os conjuntos sao iguais
11- Ver se um conjunto eh vazio
12- Imprimir conjunto
0- Sair
12

Escolha qual conjunto deseja imprimir:
1- Conjunto A
2- Conjunto B
2
Conjunto: (4,7,9)

Aperte qualquer tecla para continuar
█
```

Figura 6: Questão 1.2 - Conjunto B impresso na tela

```
Digite o que deseja fazer:
1- Inserir um numero no conjunto
2- Remover um numero no conjunto
3- Ver o tamanho do conjunto
4- Ver se um numero pertence ao conjunto
5- Ver qual o maior numero do conjunto
6- Ver qual o menor numero do conjunto
7- Ver a uniao entre dois conjuntos
8- Ver a interseccao entre dois conjuntos
9- Ver a diferenca entre dois conjuntos
10- Ver se os conjuntos sao iguais
11- Ver se um conjunto eh vazio
12- Imprimir conjunto
0- Sair
2

Escolha qual conjunto deseja fazer a remocao:
1- Conjunto A
2- Conjunto B
1

Digite o numero que deseja remover do conjunto A: 6 █
```

Figura 7: Questão 1.2 - Uso da função *removeElemento*

```

2- Remover um numero no conjunto
3- Ver o tamanho do conjunto
4- Ver se um numero pertence ao conjunto
5- Ver qual o maior numero do conjunto
6- Ver qual o menor numero do conjunto
7- Ver a uniao entre dois conjuntos
8- Ver a interseccao entre dois conjuntos
9- Ver a diferenca entre dois conjuntos
10- Ver se os conjuntos sao iguais
11- Ver se um conjunto eh vazio
12- Imprimir conjunto
0- Sair
12

Escolha qual conjunto deseja imprimir:
1- Conjunto A
2- Conjunto B
1
Conjunto: (5,7)

Aperte qualquer tecla para continuar

```

Figura 8: Questão 1.2 - Conjunto A impresso na tela

```

2- Remover um numero no conjunto
3- Ver o tamanho do conjunto
4- Ver se um numero pertence ao conjunto
5- Ver qual o maior numero do conjunto
6- Ver qual o menor numero do conjunto
7- Ver a uniao entre dois conjuntos
8- Ver a interseccao entre dois conjuntos
9- Ver a diferenca entre dois conjuntos
10- Ver se os conjuntos sao iguais
11- Ver se um conjunto eh vazio
12- Imprimir conjunto
0- Sair
3

Escolha qual conjunto deseja fazer ver o tamanho:
1- Conjunto A
2- Conjunto B
1

O tamanho do conjunto A eh: 2

Aperte qualquer tecla para continuar

```

Figura 9: Questão 1.2 - Uso da função *tamanho*

```
2- Remover um numero no conjunto
3- Ver o tamanho do conjunto
4- Ver se um numero pertence ao conjunto
5- Ver qual o maior numero do conjunto
6- Ver qual o menor numero do conjunto
7- Ver a uniao entre dois conjuntos
8- Ver a interseccao entre dois conjuntos
9- Ver a diferenca entre dois conjuntos
10- Ver se os conjuntos sao iguais
11- Ver se um conjunto eh vazio
12- Imprimir conjunto
0- Sair
5

Escolha qual conjunto deseja saber o maior elemento
1- Conjunto A
2- Conjunto B
2

O maior elemento do conjunto B eh: 9

Aperte qualquer tecla para continuar
```

Figura 10: Questão 1.2 - Uso da função *maior*

```
2- Remover um numero no conjunto
3- Ver o tamanho do conjunto
4- Ver se um numero pertence ao conjunto
5- Ver qual o maior numero do conjunto
6- Ver qual o menor numero do conjunto
7- Ver a uniao entre dois conjuntos
8- Ver a interseccao entre dois conjuntos
9- Ver a diferenca entre dois conjuntos
10- Ver se os conjuntos sao iguais
11- Ver se um conjunto eh vazio
12- Imprimir conjunto
0- Sair
6

Escolha qual conjunto deseja saber o menor elemento
1- Conjunto A
2- Conjunto B
1

O menor elemento do conjunto A eh: 5

Aperte qualquer tecla para continuar
```

Figura 11: Questão 1.2 - Uso da função *menor*


```
Digite o que deseja fazer:
1- Inserir um numero no conjunto
2- Remover um numero no conjunto
3- Ver o tamanho do conjunto
4- Ver se um numero pertence ao conjunto
5- Ver qual o maior numero do conjunto
6- Ver qual o menor numero do conjunto
7- Ver a uniao entre dois conjuntos
8- Ver a interseccao entre dois conjuntos
9- Ver a diferenca entre dois conjuntos
10- Ver se os conjuntos sao iguais
11- Ver se um conjunto eh vazio
12- Imprimir conjunto
0- Sair
7
Conjunto: (5,7,4,9)

Aperte qualquer tecla para continuar
█
```

Figura 12: Questão 1.2 - Uso da função *uniao*

```
Digite o que deseja fazer:
1- Inserir um numero no conjunto
2- Remover um numero no conjunto
3- Ver o tamanho do conjunto
4- Ver se um numero pertence ao conjunto
5- Ver qual o maior numero do conjunto
6- Ver qual o menor numero do conjunto
7- Ver a uniao entre dois conjuntos
8- Ver a interseccao entre dois conjuntos
9- Ver a diferenca entre dois conjuntos
10- Ver se os conjuntos sao iguais
11- Ver se um conjunto eh vazio
12- Imprimir conjunto
0- Sair
8
Conjunto: (7)

Aperte qualquer tecla para continuar
█
```

Figura 13: Questão 1.2 - Uso da função *intersecao*

```
Digite o que deseja fazer:
1- Inserir um numero no conjunto
2- Remover um numero no conjunto
3- Ver o tamanho do conjunto
4- Ver se um numero pertence ao conjunto
5- Ver qual o maior numero do conjunto
6- Ver qual o menor numero do conjunto
7- Ver a uniao entre dois conjuntos
8- Ver a interseccao entre dois conjuntos
9- Ver a diferenca entre dois conjuntos
10- Ver se os conjuntos sao iguais
11- Ver se um conjunto eh vazio
12- Imprimir conjunto
0- Sair
9
Conjunto: (4,9)

Aperte qualquer tecla para continuar
```

Figura 14: Questão 1.2 - Uso da função *diferenca*

```
Digite o que deseja fazer:
1- Inserir um numero no conjunto
2- Remover um numero no conjunto
3- Ver o tamanho do conjunto
4- Ver se um numero pertence ao conjunto
5- Ver qual o maior numero do conjunto
6- Ver qual o menor numero do conjunto
7- Ver a uniao entre dois conjuntos
8- Ver a interseccao entre dois conjuntos
9- Ver a diferenca entre dois conjuntos
10- Ver se os conjuntos sao iguais
11- Ver se um conjunto eh vazio
12- Imprimir conjunto
0- Sair
10

Os conjuntos nao sao iguais

Aperte qualquer tecla para continuar
```

Figura 15: Questão 1.2 - Uso da função *ehigual*

```
2- Remover um numero no conjunto
3- Ver o tamanho do conjunto
4- Ver se um numero pertence ao conjunto
5- Ver qual o maior numero do conjunto
6- Ver qual o menor numero do conjunto
7- Ver a uniao entre dois conjuntos
8- Ver a interseccao entre dois conjuntos
9- Ver a diferenca entre dois conjuntos
10- Ver se os conjuntos sao iguais
11- Ver se um conjunto eh vazio
12- Imprimir conjunto
0- Sair
11

Escolha qual conjunto deseja saber se eh vaz
1- Conjunto A
2- Conjunto B
2

0 conjunto B nao eh vazio

Aperte qualquer tecla para continuar
```

Figura 16: Questão 1.2 - Uso da função *ehvazio*