**Introducão**

Para este trabalho foi solicitado que programasse um conjunto de todas as funcões para operar com eficácia alguns conjuntos, para fazer isto, usei TAD’s (Tipos Abstratos de Dados). As funcões implementadas foram as seguintes :

* Inicializa
* Insere
* Exclui
* Seta elemento
* Get elemento
* Testa elemento
* Gera conjunto
* Num2conj
* Conj2num
* Compara
* Imprime
* Uniao
* Fusao
* Subtracao

Alem disso, por fim foi pedido para que fosse criada outro metodo main, um jogo de adivinhacao.

**Implementacao**

**Main-** Dada pelo professor, nada mais faz do que uma sequencia de chamadas das funcoes criadas

**Inicializar-** Funcao recebe como parametro um conjunto (a ser inicializado) e não tem retorno. Ela aloca 10 posicoes para seu conjunto e o deixa com posicoes nulas

**Uniao-** Cria novo conjunto, aloca espaco para ele, e nele armazena elementos dos dois conjuntos passados por parametro.

**Testa Elemento –** Recebe como parametro um conjunto e um elemento, e retorna1 se o elemento se encontra no conjunto, e -1 se não se encontra

**Num2conj-** Recebe um interiro como parametro e o transforma num conjunto atraves de uma divisao por numeros da base 10, tirando assim o numero da esquerda, depois o segundo da esquerda, e assim por diante

**SetElemento-** Recebe como parametro um numero a posicao e o conjunto, e se o conjunto tiver essa posicao livre, o elemento podera ser incluido.

**GetElemento-** Recebe como parametro um numero, a posicao e o conjunto, efetua uma pesquisa e se o elemento se encontrar nesse conjunto, a funão o exibira

**Comparar-** Recebe dois conjuntos, percorre os dois atraves de uma estrutuda de repeticao, e se, e apenas se, todos os elementos forem iguais, retorna 1, o que significa que os conjuntos são iguais.

**Gerar Conjunto-** Recebe como parametro um único inteiro, e cria um conjunto, com valores aleatorios, do tamanho do n recebido.

**Conj2num-** Recebe um conjunto, e atraves de operacoes de potenciacao, o transforma em um numero inteiro e o retorna pra funcao que o chamou .

**Excluir-** Recebe como parametro um conjunto e um elemento, busca no conjunto a incidencia deste elemento, e se encontrar, retira-o colocando na posicao um valor nulo

**Intercecao-** Por meio de for’s encadeados, passa por todas as posicoes do conjunto, buscando elementos comuns entre si, quando encontrados estes elementos vao para um terceiro conjunto

**Subtracao-** Trabalha de forma semelhante à funcao anterior, descobre-se quao dos cinjuntos é maior, e assim subtrai-se as posicoes iguais, fazendo assim a subtracao dos conjuntos.

**Insere-** primeiramente verifica se há espaco para inserir, se sim, verifica-se se o elemento já foi inserido, e caso contrario insere-se, caso não haja espaco para armazenamento, realoca-se memoria, expandindo o conjunto em mais 10 posicoes.

**Jogo-** Primeiramente o switch encaminha o usuario ao modo de jogo que ele deseja, após isso, entra-se num while, que rodará enquando o usuario não acretar o numero gerado aleatoriamente e setado usando a funcao insere\_elemento, duas funcoes de teste foram criadas para responder o usuario em suas tentativas, uma conta quantas posicoes e elementos certos o usuario respondeu, a outra conta quantos elementos estao certos, porem na posicao errada.

**Estudo de Complexidade**

Para o estudo analisei o pior caso

* Inicializa O(1);
* Insere O(n);
* Exclui O(n);
* Seta elemento O(n);
* Get elemento O(n);
* Testa elemento O(n);
* Gera conjunto O(1);
* Num2conj O(1)
* Conj2num O(1);
* Compara O(n);
* Imprime O(1);
* Uniao O(n);
* Fusao O(n);
* Subtracao O(n);

**Listagem de testes executados**

Os testes foram realizados de tres formas, primeiramente utilizando um menu criado por mim :

printf("\t\t\*\*\*Bem Vindo ao Programa dos conjuntos\*\*\*\n");

printf("\n\ que deseja fazer ?\n\n");

printf("1-Inicializar conjunto\n");

printf("2-Inserir Elemento\n");

printf("3-Excluir Elemento\n");

printf("4-Setar Elemento\n");

printf("5-Obter Elemento\n");

printf("6-Testar Elemento\n");

printf("7-Gerar conjunto\n");

printf("8-Criar conjunto a partir de um inteiro\n");

printf("9-Gerar inteiro a partir de conjunto\n");

printf("10-comparar conjuntos\n");

printf("11-Imprimir conjunto\n");

printf("12-Uniao de conjuntos\n");

printf("13-Intersecao de conjuntos\n");

printf("14-Subtrair conjuntos\n");

scanf("%d",&menu);

após todos os testes, testei a aplicacao com o main fornecido pelo professor,e depois com o jogo dos acertos .

**Conclusão**

Ao fazer o trabalho, pude expandir meus conhecimentos em programacao, já não usarei estruturas convencionais, visto a facilidade e o leque de possibilidades que o uso de TADs podem trazer, sem falar da organizacao dos dados.