

Relatório

Trabalho 02 – LPA

- Introdução: Este relatório refere-se ao segundo trabalho da disciplina SSC0300 – Linguagens de Programação e Aplicações, referente à operações com árvores binárias de busca.
Os integrantes do grupo são :
1 - Hermano Esch Ferreira da Costa N°USP 9312710
2 - Rodrigo Anunciação N°USP 9312706
3 - Luís Adolfo Mazini RodriguesN°USP8125571
4 - Afonso Sorci Ferreira N°USP9312561
5 – Pedro Eduardo Rodrigues BarazettiN°USP 9312582
- Descrição do Projeto:Os códigos nesse repositório foram criados em linguagem C(.c) utilizando o DEV-C++ 5.10 em Windows 8.1 64 bits O compilador usado na compilação dos códigos foi o TDM-GCC 4.8.1 64-bit Release. O primeiro exercício utiliza as bibliotecas stdio.h e stdlib.h. O segundo exercício utiliza as bibliotecas stdlib.h, stdio.h e string.h(apenas no exercício 2) para sua execução.

Exercícios:

1 - Operações Básicas em Árvores:

Este código recebe inicialmente uma raiz, que inicializa a árvore binária de busca(ABB). Então é exibido um menu, com as opções de inserir um novo elemento na árvore, remover um elemento, e funções de impressão(em ordem, pós ordem, pré ordem e labelledbracketing).

A árvore é organizada com a partir da raiz inserida inicialmente, com os números menores à esquerda da raiz, e os maiores à direita da raiz.

Entrada exemplo:

```
3
6
8
1
      15
9
      20
```

Onde a raiz inserida foi o 3.

- Tutorial 1: Para um bom funcionamento do programa ele deve ser compilado de acordo com o abordado na descrição do projeto acima. Primeiramente será pedido a raiz a ser inserida. Após isso um menu aparecerá, onde através dele é possível executar qualquer operação descrita acima na árvore. Enquanto o numero “5” não for entrado no menu(corresponde à função exit), o menu continuará a aparecer após cada operação realizada pelo executante.

A imagem a seguir mostra o funcionamento do programa:

```

C:\Users\Rodrigo Anunciap\Oo\Desktop\Trabalho 02 -LPA\Trabalho-02---LPA\E...
Digite a raiz: 3
Digite 1 para inserir, 2 para remover, 3 para buscar, 4 para imprimir, 5 para sair: 1
Digite o numero de elementos a serem inseridos: 6
Digite 1-o numero a ser inserido: 6
6 - dir do 3
Digite 2-o numero a ser inserido: 8
8 - dir do 6
Digite 3-o numero a ser inserido: 1
1 - esq do 3
Digite 4-o numero a ser inserido: 15
15 - dir do 8
Digite 5-o numero a ser inserido: 9
9 - esq do 15
Digite 6-o numero a ser inserido: 20
20 - dir do 15
Digite 1 para inserir, 2 para remover, 3 para buscar, 4 para imprimir, 5 para sair: 2
Digite o elemento a ser removido: 8
Digite 1 para inserir, 2 para remover, 3 para buscar, 4 para imprimir, 5 para sair: 3
Digite o elemento a ser buscado: 1
0 numero esta na arvore!
Digite 1 para inserir, 2 para remover, 3 para buscar, 4 para imprimir, 5 para sair: 4
Digite 1 para em ordem, 2 para preordem, 3 para posordem, 4 para labelled: 1
1
3
6
9
15
20
Digite 1 para inserir, 2 para remover, 3 para buscar, 4 para imprimir, 5 para sair: 5

-----
Process exited after 55.27 seconds with return value 4201212
Pressione qualquer tecla para continuar. . .

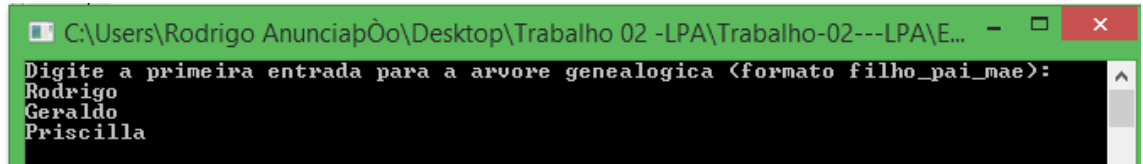
```

2 – Árvore Genealógica:

Este código recebe primeiramente uma entrada do tipo filho_pai_mae, inicializando o programa. Após inserido a primeira tripla, um menu é exibido, com as opções:

- 1 -Inserir: Insere mais triplas filho_pai_mae na árvore.
- 2 – Grau e Parentesco: retorna o grau de parentesco entre duas pessoas dadas.
- 3 – Imprimir: essa opção permite ao usuário imprimir a árvore por geração, imprimir em labelledbracketing, e também imprimir os antepassados de um indivíduo.
- 4 – Sair: sai do programa.

Entrada do programa:



```
C:\Users\Rodrigo AnunciabOo\Desktop\Trabalho 02 -LPA\Trabalho-02---LPA\E... - □ ×
Digite a primeira entrada para a arvore genealogica <formato filho_pai_mae>:
Rodrigo
Geraldo
Priscilla
```

- Tutorial 2: Para um bom funcionamento do programa ele deve ser compilado de acordo com o abordado na descrição do projeto acima.

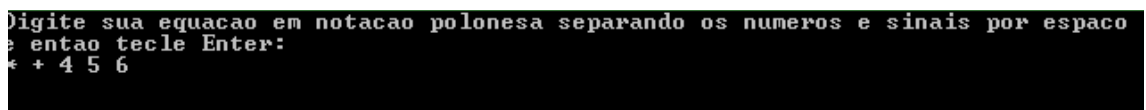
É considerado apenas um filho para cada casal. Primeiramente será pedida uma entrada inicial de tripla de nomes na árvore, do modo filho_pai_mae. Após feito, será exibido um menu, com as 4 opções descritas acima. O código foi feito através de funções separadas:

```
struct arvore * search;
void Insertion;
int Parentesco;
void ImpGeracaoN;
void ImpGeracaoTotal;
void ImpLabel;
```

Todas essas funções são parte da main, onde são possíveis realizar todas as opções do menu (feito com switch). Para escolher uma opção, deve-se digitar o número correspondente do menu, e teclar Enter. A cada dado inserido no programa, deve-se teclar Enter para prosseguir.

3 – Conversor de equação Polonesa:

Este código recebe uma equação do tipo “equação polonesa”, dessa maneira:



```
Digite sua equacao em notacao polonesa separando os numeros e sinais por espaco
e entao tecle Enter:
* + 4 5 6
```

Com as operações anteriormente aos números, e então é exibido um menu com as opções de representar a equação escrita em RPN ou em notação infixa ou sair do programa, por exemplo, a infixa a seguir é ilustrada:

```

Digite sua equacao em notacao polonesa separando os numeros e sinais por espaco
e entao tecle Enter:
* + 4 5 6
Digite 1 para imprimir em RPN, 2 para imprimir em notacao infixa e 3 para sair
2
Infixa:
(4)+(5)*(6)
Digite 1 para imprimir em RPN, 2 para imprimir em notacao infixa e 3
para sair

```

- Tutorial 3: Para um bom funcionamento do programa ele deve ser compilado de acordo com o abordado na descrição do projeto acima.

Deve-se digitar os sinais operadores à esquerda, separando cada sinal/numero por um espaço, e ao fim da equação, teclar Enter para abrir o menu.

Após fazer uma escolha do menu, o menu retorna, podendo o usuário imprimir a equação em RPN e em Infixa sem precisar sair do programa, quando digitado o numero 3 no menu, o programa é parado.