

Trabalho de Linguagens de Programação e Aplicações

1. Introdução

Este trabalho tem como principais objetivos demonstrar as aplicações práticas do conteúdo aprendido em sala de aula. Conceitos como recursão, árvores e ordenação de dados foram amplamente utilizados neste a fim de comprovar o nosso aprendizado. É importante frisar que para melhor organização e eficiência dos dados os membros do grupo aprenderam um conceito muito utilizado em Linguagem C: a criação de bibliotecas- o que pode ser observado nos códigos dos programas.

Participaram da criação do trabalho os alunos: Flávio Sarti (9312578), Vitor Eiti Uekawa (9012671), Elizabete Vindilino Santos (9312922), Érico Tadeu de Paula Vieira (9312727) e Felipe Caron Figueiredo (9311664) os quais devem receber a pontuação de acordo com a ordem apresentada. É importante frisar que todos contribuíram para a criação do trabalho, sendo a ordem de pontuação escolhida de maneira aleatória pelos membros do grupo.

O Trabalho 2 pode ser dividido em três exercícios: Operações Básicas em Árvores, Árvore genealógica e Conversor de Equação Polonesa. Tais exercícios são melhor elucidados a seguir.

2. Descrição do projeto

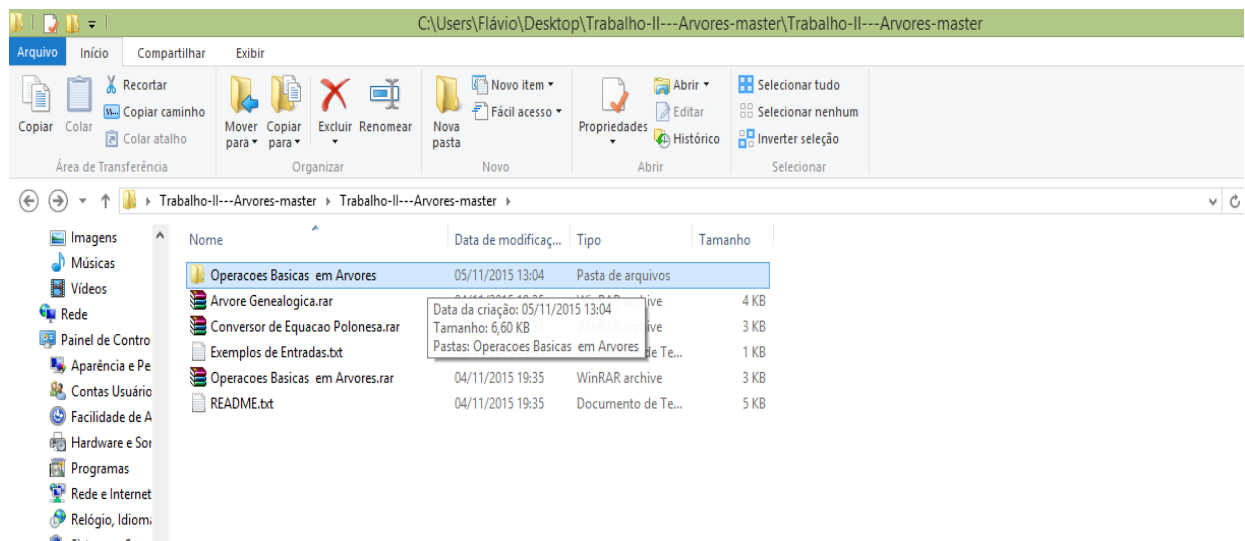
A seguir segue uma descrição sucinta do projeto:

- | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ul style="list-style-type: none">2.1- Linguagem: C2.2- Ambiente de desenvolvimento: Code::Blocks 13.122.3-Parâmetros de compilação: padrão do ambiente2.4-Arquitetura: Windows 8.1 64 bits2.5-Compilador: mingw32-gcc.exe - GNU GCC Compiler2.6-Bibliotecas: stdio.h, stdlib.h, abb.h, arvore.h, arvoreGenealogica.h; sendo as últimas três criadas2.7-Exemplos de entradas: os exemplos de entradas encontram-se em um arquivo .txt presente no diretório |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

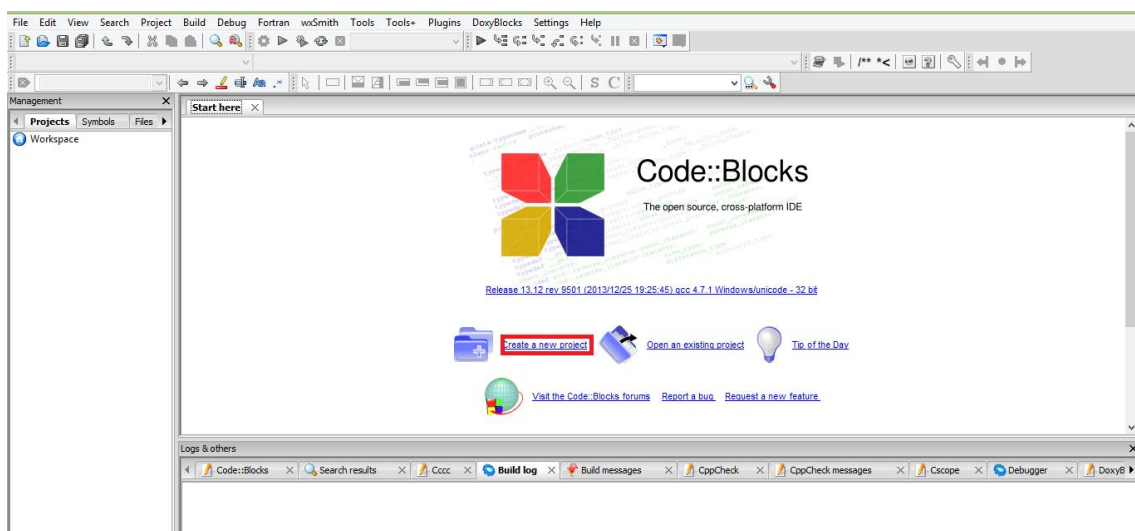
3. Tutorial

Esta seção tem como principal objetivo fornecer ao usuário um manual de como executar de forma eficiente os programas.

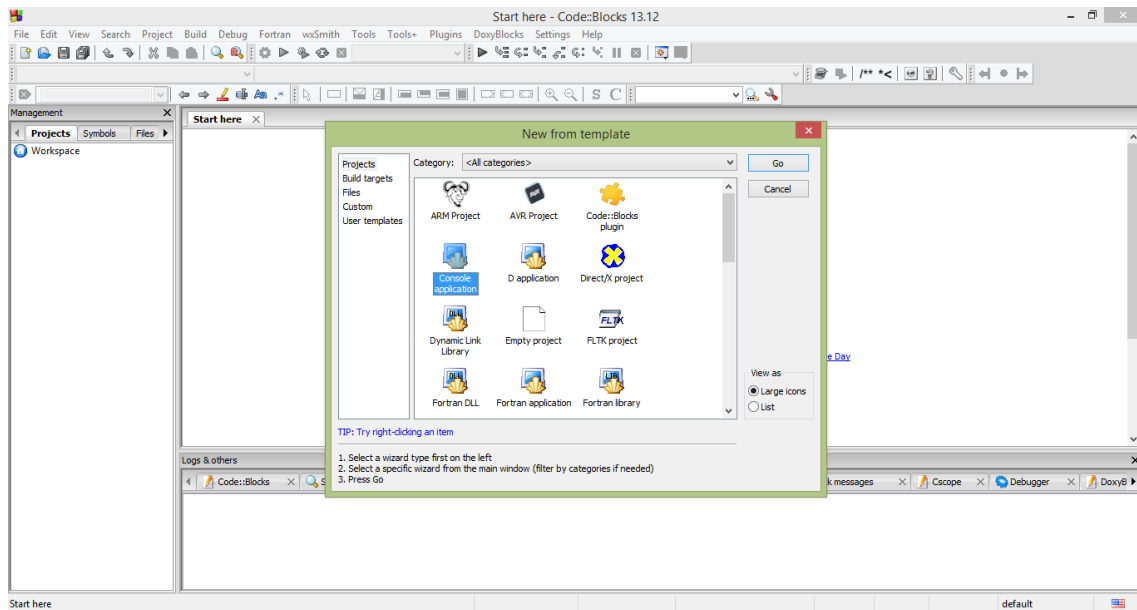
Após baixar os arquivos do diretório do GitHub em arquivo .zip, descompacte o programa a ser executado:



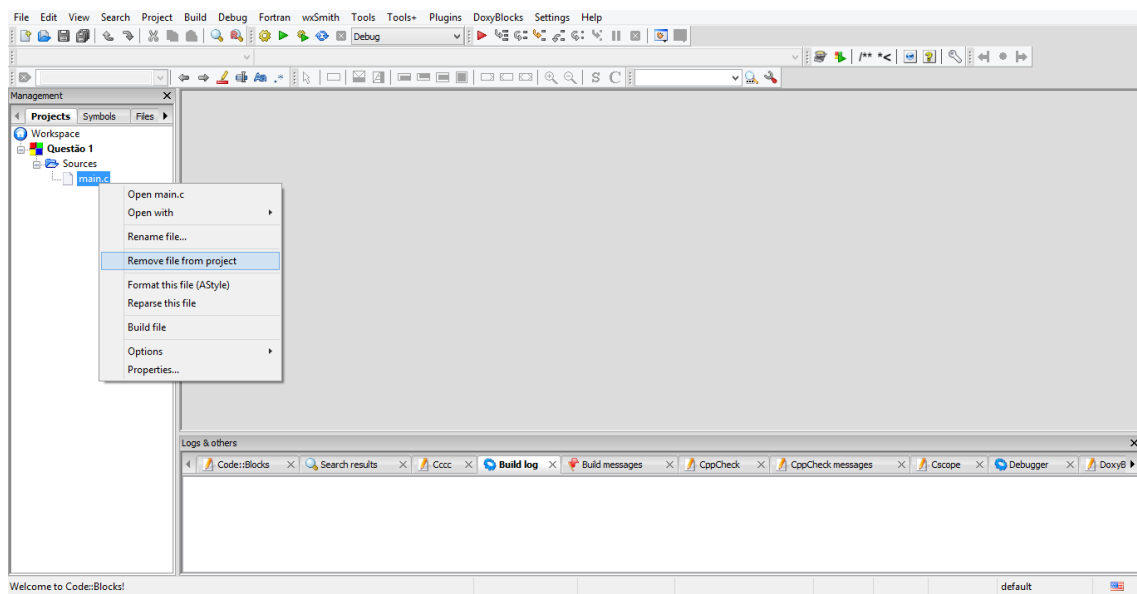
Abra o “Code Blocks” e selecione a opção “Criar novo projeto”:



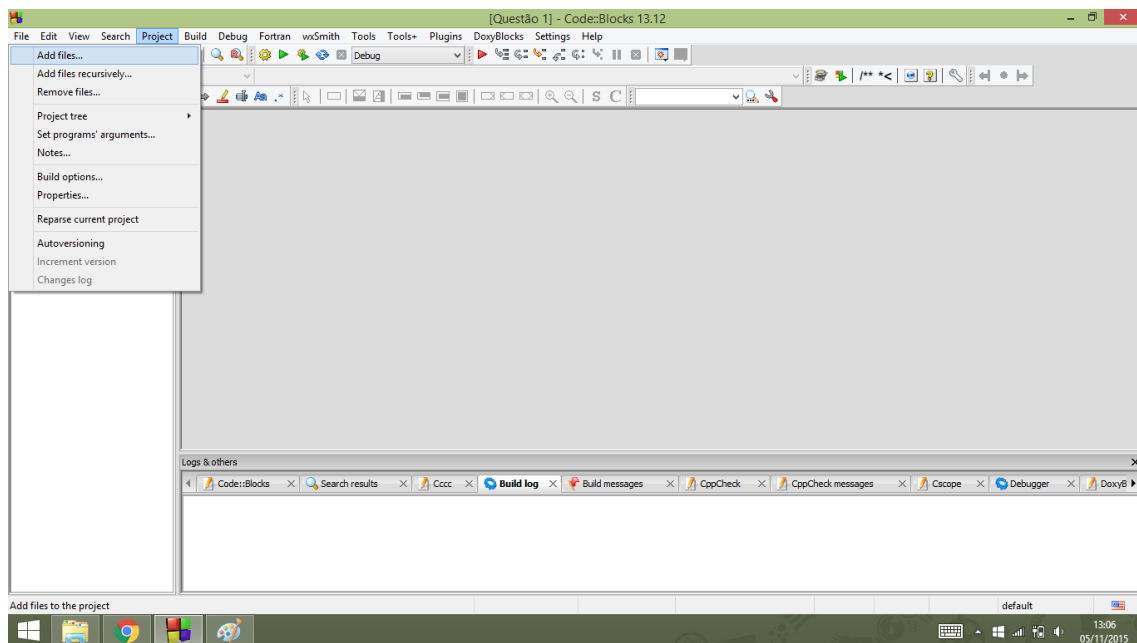
Selecione a opção “Console application”:



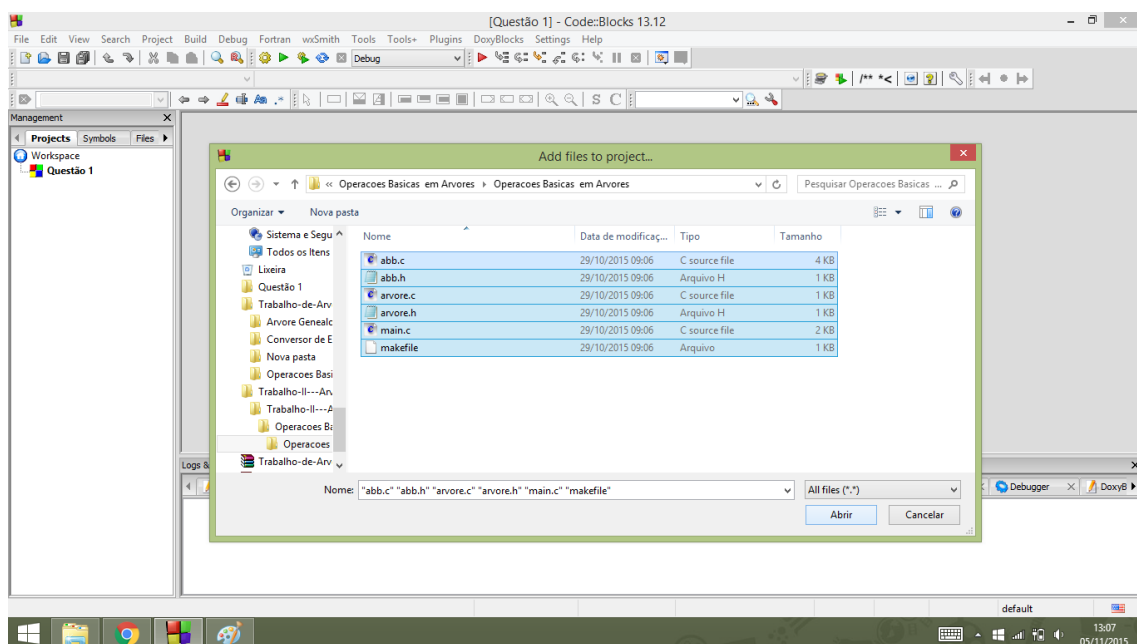
Delete a função “main.c” já presente no Code Blocks:



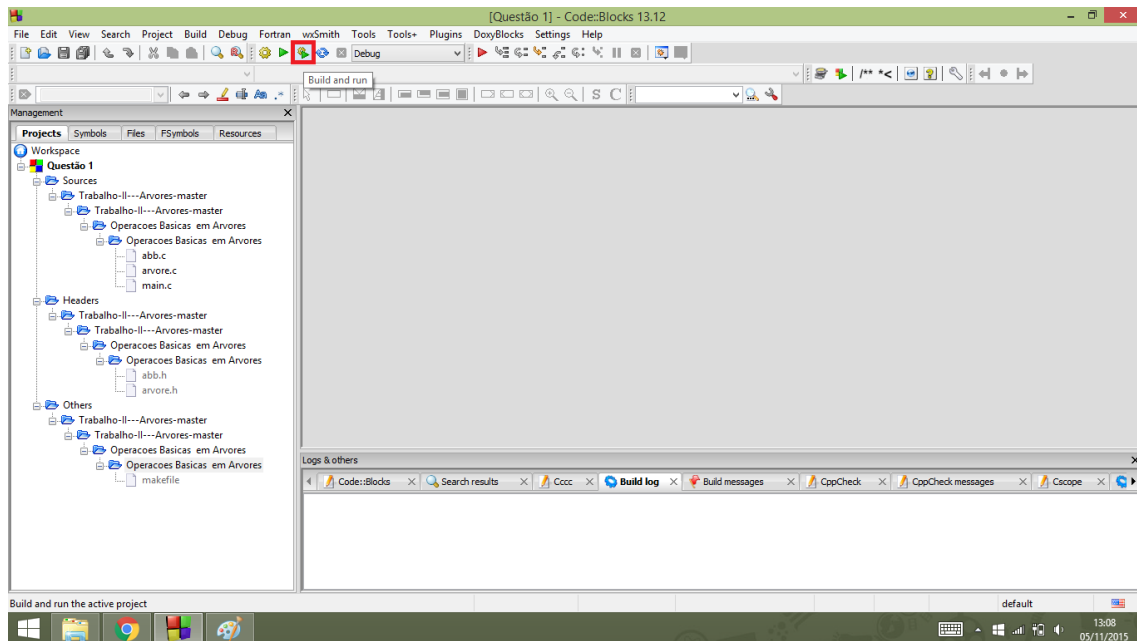
Adicione os arquivos da pasta



Selecione todos os arquivos da pasta descompactada no primeiro passo:



Selecione a opção “Build and run”:



Pronto ! O programa está pronto para ser executado!

a) Operações Básicas em Árvores: o programa atrelado a esse exercício tem como principal objetivo armazenar números em uma árvore binária e possibilitar ao leitor a escolha do tipo de manipulação a ser feita na árvore (ex.: buscar, remover e imprimir elementos da árvore).

Após inseridos os dados (números) na árvore o usuário deve, por meio de um menu, selecionar a o tipo de operação que deseja realizar na árvore.

```

1 - Inserir elemento
2 - Remover elemento
3 - Buscar elemento
4 - Impressão Em Ordem
5 - Impressão Pré Ordem
6 - Impressão Pós Ordem
7 - Impressão Labelled Bracketing
8 - Sair
>

```

É importante destacar que a cada entrada fornecida o usuário deve pressionar a tecla “Enter”.

Dependendo da opção escolhido no menu o usuário receberá uma mensagem que pergunte por exemplo qual valor deseja remover (caso o valor não esteja na árvore o usuário receberá uma mensagem) ou qual valor deseja ser buscado (o usuário receberá uma mensagem avisando se o valor está ou não na árvore).

```
0 - Sair
> 1
Digite o valor a ser inserido > 5
Valor inserido com sucesso

1 - Inserir elemento
2 - Remover elemento
3 - Buscar elemento
4 - Impressão Em Ordem
5 - Impressão Pré-Ordem
6 - Impressão Pós-Ordem
7 - Impressão Labelled Bracketing
0 - Sair
> 2
Digite o valor a ser removido > 4
Valor não encontrado na árvore

1 - Inserir elemento
2 - Remover elemento
3 - Buscar elemento
4 - Impressão Em Ordem
5 - Impressão Pré-Ordem
6 - Impressão Pós-Ordem
7 - Impressão Labelled Bracketing
0 - Sair
```

Caso o usuário selecione uma opção relacionada a impressão dos dados da árvore o programa imprimirá esses de acordo com a opção selecionada (Pré-ordem, pós ordem ou “labelled bracketing”).

```

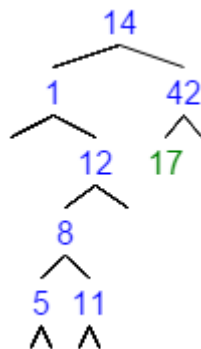
0 - Sair
> 1
Digite o valor a ser inserido > 8
Valor inserido com sucesso

1 - Inserir elemento
2 - Remover elemento
3 - Buscar elemento
4 - Impressão Em Ordem
5 - Impressão Pré-Ordem
6 - Impressão Pós-Ordem
7 - Impressão Labelled Bracketing
0 - Sair
> 5
5 1 3 6 9 8

1 - Inserir elemento
2 - Remover elemento
3 - Buscar elemento
4 - Impressão Em Ordem
5 - Impressão Pré-Ordem
6 - Impressão Pós-Ordem
7 - Impressão Labelled Bracketing
0 - Sair
>

```

Após realizar todas as operações a árvore é destruída sendo a memória utilizada “desalocada” para melhor proveito do espaço disponível. A seguir uma imagem da árvore binária:



b) Árvore genealógica:

O usuário deverá primeiramente fornecer o número de gerações que deseja analisar. Posteriormente deve fornecer os nomes dos membros da família (separados por espaço) que tiverem os seguintes parentescos “filho, pai e mãe”. O usuário deve sempre fornecer a tupla de relações nessa ordem.

```

4
isabel pedroII thereza
pedroII pedroI leopoldina
leopoldina franciscoII teresaBourbon
pedroI joaoVI carlota
1 - Imprimir membros da família
2 - Imprimir antepassados
3 - Imprimir árvore usando Labelled Bracketing
4 - Calcular grau de parentesco
0 - Sair
>

```

Após fornecer os membros de todas as gerações o usuário deve pressionar a tecla “ENTER”. Aparecerá um menu e por meio desse o usuário deve selecionar o tipo de operação que deseja realizar na árvore (imprimir membros da família, imprimir antepassados, imprimir árvore usando “labelled bracketing”, calcular grau de parentesco ou sair).

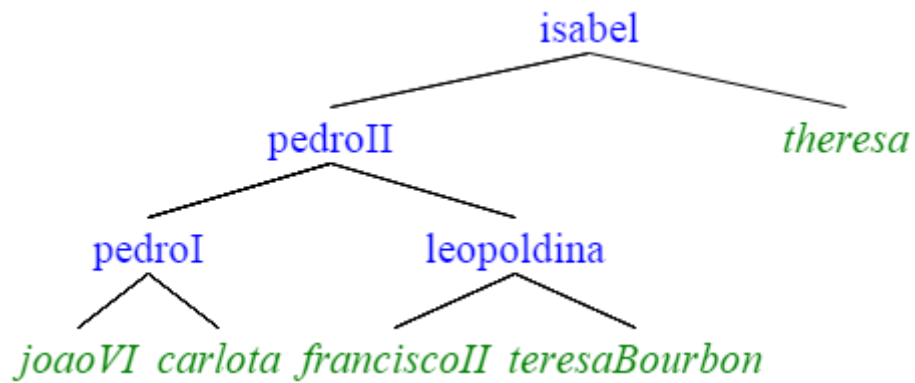
De acordo com a opção selecionada o programa fornecerá os dados solicitados:

```

4
isabel pedroII thereza
pedroII pedroI leopoldina
leopoldina franciscoII teresaBourbon
pedroI joaoVI carlota
1 - Imprimir membros da família
2 - Imprimir antepassados
3 - Imprimir árvore usando Labelled Bracketing
4 - Calcular grau de parentesco
0 - Sair
> 3
[isabel [pedroII [pedroI [joaoVI [] []] [carlota [] []]] [leopoldina [franciscoII
I [] []] [teresaBourbon [] []]]] [thereza [] []]]
1 - Imprimir membros da família
2 - Imprimir antepassados
3 - Imprimir árvore usando Labelled Bracketing
4 - Calcular grau de parentesco
0 - Sair
>

```

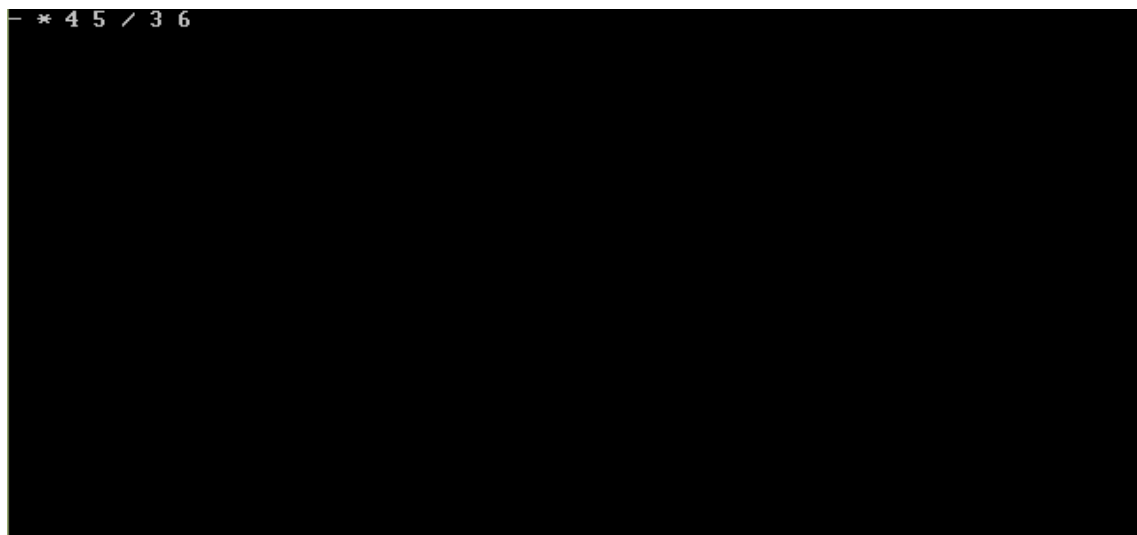

Ao selecionar a opção “0-sair” a árvore é destruída e a memória desalojada para otimização do espaço disponível. É importante destacar que nos programas atrelados às opções selecionadas são usados recursos como recursão a fim de economia da memória disponível. A seguir uma imagem da árvore binária:



c) Conversor de Equação polonesa

Este exercício tem como objetivo realizar conversões de notações matemáticas. O programa recebe do usuário uma equação em notação polonesa e a converte para notação polonesa reversa e infixa.

Primeiramente o usuário deve inserir a equação polonesa no executável (entre cada símbolo o usuário deve pressionar a tecla espaço).



Posteriormente o usuário deve pressionar a tecla "Enter".

```
- * 4 5 / 3 6
RPN:      4 5 * 3 6 / -
Nota|º|úo Infixa:  ((4 * 5) - (3 / 6))
Process returned 0 (0x0)   execution time : 145.045 s
Press any key to continue.
```

Assim o programa fornecerá a equação polonesa em notação infixa e RPN.0

Anexo

Neste anexo listamos algumas limitações do programa.

- b) Árvore genealógica: ao pesquisar o grau de parentesco o usuário deve primeiramente inserir o nome da pessoas mais nova. Ex.:

filho pai(CORRETO!)

pai filho (ERRADO!)