

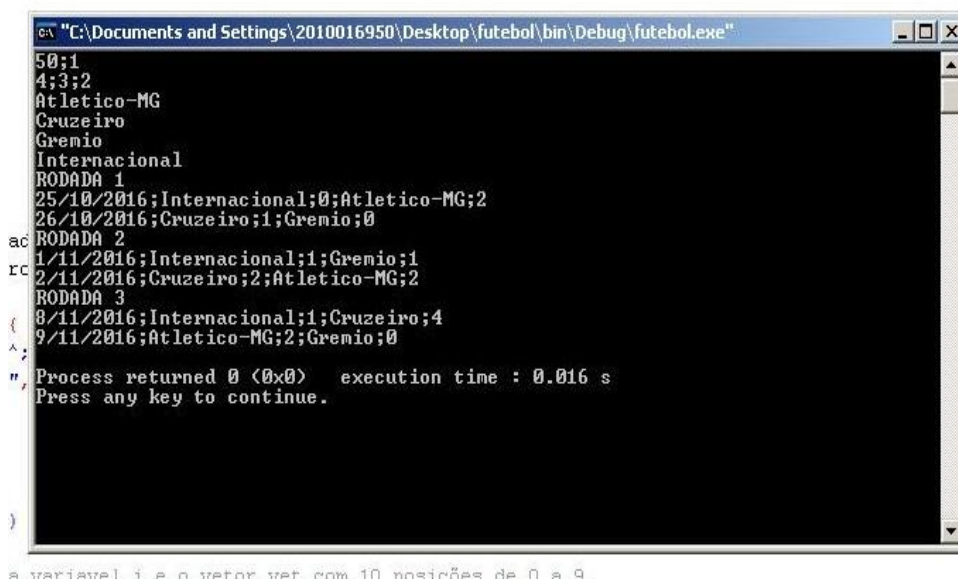
Trabalho Pratico 2

Estudante: Tarcísio de Oliveira Rocha

Nº de matricula: 2010016950

Introdução:

Testes:



```
C:\Documents and Settings\2010016950\Desktop\futbol\bin\Debug\futbol.exe
50;1
4;3;2
Atletico-MG
Cruzeiro
Gremio
Internacional
RODADA 1
25/10/2016;Internacional;0;Atletico-MG;2
26/10/2016;Cruzeiro;1;Gremio;0
RODADA 2
1/11/2016;Internacional;1;Gremio;1
2/11/2016;Cruzeiro;2;Atletico-MG;2
RODADA 3
8/11/2016;Internacional;1;Cruzeiro;4
9/11/2016;Atletico-MG;2;Gremio;0
Process returned 0 (0x0)   execution time : 0.016 s
Press any key to continue.
```

FIGURA 1: Essa figura mostra, a coleta de dados, nada mais é do que o print dos dados coletados. Esse teste sempre me ajuda, pois posso ter ideia do que estou lendo de verdade, e se estou coletado algum dado erroneamente.

```
for (j=0; j<quantidade_de_jogos; j++) {
    fscanf(entrada, "%d;%d;%d;%d;%d;%d;%d;%d;%d;%d\n", &A.c, &B.c, &C.c, &D.c, &E.c, &F.c, &G.c, &H.c, &I.c, &J.c);
    printf("%d/%d/%d;%d;%d;%d;%d;%d;%d;%d\n", A.c, B.c, C.c, D.c, E.c, F.c, G.c, H.c, I.c, J.c);
} // fim da for interna
} // fim da for externa

printf("\nvetor:");
int vect[10] = {2,3,6,7,8,9,10,1,5,4};

for (i=0; i<10; i++) {
    printf("%d ", vect[i]);
}
printf("\nshellsort: \n");
shellSort(vect, 10);

for (i=0; i<10; i++) {
    printf("%d ", vect[i]);
}
```



```
C:\Documents and Settings\2010016950\Desktop\futbol\bin\Debug\futbol.exe
50;1
4;3;2
Atletico-MG
Cruzeiro
Gremio
Internacional
RODADA 1
25/10/2016;Internacional;0;Atletico-MG;2
26/10/2016;Cruzeiro;1;Gremio;0
RODADA 2
1/11/2016;Internacional;1;Gremio;1
2/11/2016;Cruzeiro;2;Atletico-MG;2
RODADA 3
8/11/2016;Internacional;1;Cruzeiro;4
9/11/2016;Atletico-MG;2;Gremio;0
vetor:2 3 6 7 8 9 10 1 5 4
shellsort:
1 2 3 4 5 6 7 8 9 10
Process returned 0 (0x0)   execution time : 0.016 s
Press any key to continue.
```

FIGURA 2: como podem ver, neste teste executo o algoritmo de ordenação shellsort para um vetor de 10 posições. Ocorreu tudo como esperado.

```

printf("\nvetor:");
int vect[10] = {2,3,6,7,8,9,10,1,5,4};

for(i=0;i<10;i++){

    printf("%d ",vect[i]);

}

/// PASSO 2 (sobre algoritmos de Ordenação)

printf("Quicksort:");
quick(vect,0,10-1);
for(i=0;i<10;i++)
    printf("%d ",vect[i]);
printf("\n");
return 0;

```

```

50:1
4:3:2
Atletico-MG
Cruzeiro
Gremio
Internacional
RODADA 1
25/10/2016;Internacional;0;Atletico-MG;2
26/10/2016;Cruzeiro;1;Gremio;0
RODADA 2
1/11/2016;Internacional;1;Gremio;1
2/11/2016;Cruzeiro;2;Atletico-MG;2
RODADA 3
8/11/2016;Internacional;1;Cruzeiro;4
9/11/2016;Atletico-MG;2;Gremio;0
vetor:2 3 6 7 8 9 10 1 5 4 Quicksort:1 2 3 4 5 6 7 8 9 10
Process returned 0 (0x0)   execution time : 0.016 s
Press any key to continue.

```

FIGURA3: Esta figura representa a execução do algoritmo de ordenação quicksort para um vetor de 10 posições, como podem ver ocorreu tudo dentro do esperado.

número da rodada deve ser impresso em uma linha seguido da tabela de classificação com os pontos, vitórias, saldo de gols e gols marcados de cada time. Por último, os resultados para cada busca feita devem ser listados. Quando mais de um jogo ocorrer no mesmo dia, o mesmo padrão mostrado abaixo deve ser seguido, com cada jogo aparecendo em uma linha. Abaixo segue o exemplo de como deve ficar arquivo de saída:

```

1
Atlético-MG 3 1 2 2
Cruzeiro 3 1 1 1
Grêmio 0 0 -1 0
Internacional 0 0 -2 0
2
Atlético-MG 4 1 2 4
Cruzeiro 4 1 1 3
Grêmio 1 0 -1 1
Internacional 1 0 -2 1
3
Cruzeiro 7 2 4 7
Atlético-MG 7 2 4 6
Grêmio 1 0 -3 1
Internacional 1 0 -5 2
26/10/2016;Cruzeiro;1;Grêmio;0
02/11/2016;Cruzeiro;2;Atlético-MG;2

```

3.3 Compilação e Execução

O programa deverá ser possível de ser compilado sem nenhuma biblioteca que seja específica do sistema operacional windows. Descreva na sua documentação como compilar e executar seu código.

```

saida - Bloco de notas
Arquivo  Editar  Formatar  Exibir  Ajuda
1
Atlético-MG 3 1 2 2
Cruzeiro 3 1 1 1
Grêmio 0 0 -1 0
Internacional 0 0 -2 0
2
Atlético-MG 4 1 2 4
Cruzeiro 4 1 1 3
Grêmio 1 0 -1 1
Internacional 1 0 -2 1
3
Cruzeiro 7 2 4 7
Atlético-MG 7 2 4 6
Grêmio 1 0 -3 1
Internacional 1 0 -5 2

```

Figura 4. Neste passo o programa já está quase no seu fim, como podem o meu algoritmo atendeu a todos os criterios de ordenação.

Conclusão:

No início do programa dificuldade em perceber que o %s (mascara para string) estava lendo tudo até mesmo o ponto e vírgula, ocasionando em erro (40 minutos, tempo perdido com esse erro) que infelizmente não é apurado pelo compilador. No site homepages.dcc.ufmg.br, os arquivos que contem o funcionamento das arvores SBB estão errados, diferenciando-se até dos arquivos que estão no moodle da disciplina (que também não compilam), pude notar um erro na função `iinsere()`, quando ocorria a chamada desta função o programador errou ao passar por referência. O correto seria `iinsere(reg,&(*ptr)->esq,&(*ptr)->esqtipo, fim);` e não `iinsere(reg,&(*ptr)->esq,&(*ptr)->esqtipo, fim);` como está no arquivo.

Referências:

https://pt.wikipedia.org/wiki/Shell_sort#C3.B3digo_em_C acesso: segunda-feira, 21 de novembro de 2016 as 18:26.

<https://pt.wikipedia.org/wiki/Quicksort#C> acesso: segunda-feira, 21 de novembro de 2016 as 18:28.

<http://www.cprogressivo.net/2012/11/A-funcao-printf-Caracteres-Especais.html> acesso: terça-feira, 22 de novembro de 2016 as 20:33.

<http://homepages.dcc.ufmg.br/~cunha/teaching/20121/aeds2/sbbs.pdf> acesso: terça-feira, 22 de novembro de 2016 as 20:48.

https://virtual.ufmg.br/20162/pluginfile.php/250420/mod_resource/content/1/aeds2_aula_0_20_hashing-fab.pdf acesso: terça-feira, 22 de novembro de 2016 as 21:29.

<https://www.ime.usp.br/~pf/algoritmos/aulas/stru.html> acesso : quarta-feira, 23 de novembro de 2016 as 17:33.

<https://www.ime.usp.br/~pf/mac0122-2002/aulas/ hashing.html> acesso : quarta-feira, 23 de novembro de 2016 as 18:44.

<https://tentandoblogar.wordpress.com/2009/03/15/comparando-palavras-em-c-a-funcao-strcmp/> acesso: quarta-feira, 23 de novembro de 2016 as 21:23.

<http://www.cprogressivo.net/2013/09/Header-cabecalho-o-que-sao-para-que-servem-como-criar-e-usar-seus-arquivos-.h.html> acesso: segunda-feira, 23 de novembro de 2016 as 20:17.

<http://www.guj.com.br/t/resolvido-como-comparar-strings-em-c/283211> acesso: segunda-feira, 23 de novembro de 2016 as 21:37

