with Ada.Text\_IO; use Ada.Text\_IO;

with Ada.Directories; use Ada.Directories;

with GNAT.Regexp; use GNAT.Regexp;

with Ada.Strings.Unbounded; use Ada.Strings.Unbounded;

with Ada.Integer\_Text\_IO; use Ada.Integer\_Text\_IO;

with Ada.Strings.Fixed; use Ada.Strings.Fixed;

with Ada.Text\_IO, ADA.Float\_Text\_IO,ADA.Integer\_Text\_IO; use Ada;

procedure Main is

type Decimal is delta 0.01 digits 10;

F : File\_Type;

pathDir : String(1..100);

pathLen : Integer;

procedure Search\_Files (Dir : String; Pattern : String) is

Search : Search\_Type;

Ent : Directory\_Entry\_Type;

Re : constant Regexp := Compile (Pattern, Glob => True);

SumBuy : Float := 0.0;

SumSell: Float := 0.0;

TempFloat : Float;

infp : File\_Type;

sline : string(1..100);

lineLength : Natural;

indexArray : array (1..3) of Integer;

nextIndex : Natural;

function HandleBuy (ProductName : String; Amount : Integer; Price : Float) return Float is

begin

Put\_Line(F, "### Buy " &ProductName& " ###");

TempFloat := Float(Amount) \* Price;

Float\_Text\_IO.Put(F,TempFloat, Aft=>1, Exp=>0);

Put\_Line(F,"");

return TempFloat;

end HandleBuy;

function HandleSell (ProductName : String; Amount : Integer; Price : Float) return Float is

begin

Put\_Line(F, "### Sell " &ProductName& " ###");

TempFloat := Float(Amount) \* Price;

Float\_Text\_IO.Put(F,TempFloat, Aft=>1, Exp=>0);

Put\_Line(F,"");

return TempFloat;

end HandleSell;

begin

Start\_Search (Search, Directory => Dir, Pattern => "");

while More\_Entries (Search) loop

Get\_Next\_Entry (Search, Ent);

if Match (Simple\_Name (Ent), Re) then

Put\_Line(F, Simple\_Name(Ent)(1..Simple\_Name(Ent)'Length-3));

open(infp, in\_file, Dir&"\"&Simple\_Name(Ent));

loop

exit when end\_of\_file(infp);

get\_line(infp, sline, lineLength);

nextIndex := 1;

for i in 1..3 loop

nextIndex := Index(Source => sline(1..lineLength),

Pattern => " ",

From => nextIndex+1);

indexArray(i) := nextIndex;

end loop;

if sline(1..3) = "buy" then

SumBuy := SumBuy + HandleBuy(ProductName => sline(indexArray(1)+1..indexArray(2)),

Amount => Integer'Value(sline(indexArray(2)+1..indexArray(3))),

Price => Float'Value(sline(indexArray(3)+1..lineLength)));

else

SumSell := SumSell + HandleSell(ProductName => sline(indexArray(1)+1..indexArray(2)),

Amount => Integer'Value(sline(indexArray(2)+1..indexArray(3))),

Price => Float'Value(sline(indexArray(3)+1..lineLength)));

end if;

end loop;

Close(infp);

end if;

end loop;

Put(F,"TOTAL BUY: ");

Float\_Text\_IO.Put(F,SumBuy, Aft=>1, Exp=>0);

Put\_Line(F,"");

Put(F,"TOTAL SELL: ");

Float\_Text\_IO.Put(F,SumSell, Aft=>1, Exp=>0);

Put\_Line(F,"");

Put("TOTAL BUY: ");

Float\_Text\_IO.Put(SumBuy, Aft=>1, Exp=>0);

Put\_Line("");

Put("TOTAL SELL: ");

Float\_Text\_IO.Put(SumSell, Aft=>1, Exp=>0);

Put\_Line("");

End\_Search (Search);

end Search\_Files;

begin

Get\_Line(pathDir, pathLen);

Create(F, Out\_File, pathDir(1..pathLen)&".asm");

Search\_Files (pathDir(1..pathLen), "\*.vm");

Close(F);

end Main;