

Lab 1 A

Programming language docs:

Rules:

used for comments:

keyword start at the beginning of the compilable program

keyword end at the end of the compilable program (anything after end is not taken anymore into account by the compiler)

= represented with "takes"

> represented with "greater"

< - represented with "smaller"

>= - represented with "greaterEq"

<= - represented with "smallerEq"

== - is

!= - isNot

read(variable) to read a value for variable

write(x) to write the content between the brackets

every statement must end with semicolon

P1

Compute maximum of 3 numbers

P1

maximum of 3 numbers

start

int nr1;

int nr2;

int nr3;

int maximum;

string text;

read (nr1);

read (nr2);

read (nr3);

maximum takes nr1;

if (nr2 greaterEq maximum)

{

maximum takes nr2;

}

elif (nr3 greaterEq maximum)

{

maximum takes nr3;

```

};

write ("max is ");
write (maximum);
end

```

P2

solution of 2nd order equation

```

start
    float a;
    float b;
    float c;
    float discriminant;
    float root1;
    float root2;

    read(a);
    read(b);
    read(c);

    discriminant takes (b * b) - (4 * a * c);

    if (discriminant smaller 0)
    {
        write("No real roots");
    }
    elif (discriminant is 0)
    {
        root1 takes -b / (2 * a);
        write("One real root: ");
        write(root1);
    }
    else
    {
        root1 takes (-b + sqrt(discriminant)) / (2 * a);
        root2 takes (-b - sqrt(discriminant)) / (2 * a);
        write("The root or roots are: ");
        write(root1);
        write(" and ");
        write(root2);
    }
};
end

```

P3

sum of n numbers

start

```
int n;  
int i;  
int current_number;  
int sum;  
sum takes 0;
```

```
read(n);
```

```
i takes 1;
```

```
while (i smallerEq n)  
{  
    read(current_number);  
    sum takes sum + current_number;  
    i takes i + 1;  
};
```

```
write("sum is ");  
write(sum);
```

end

P1ERR

start

```
int nr1;  
int nr2  
# semicolon missing  
int nr3;  
int maximum;  
string text;
```

```
read (nr1);  
read (nr2);  
read (nr3);
```

```
maximum takes nr1;
```

```
if (nr2 greaterE maximum) #no greaterE relational expression exists  
{  
    maximum takes nr2;
```

```
#closing bracket missing before elif
elif (nr3 greaterEq maximum)
{
    maximum takes nr3;
};

write ("max is ");
write (maximum);
end
```