\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Report: hw1\_1

Author: F74082086 王維瀚 <leowang3268@gmail.com>

Class: 乙班

Description:

This is the first time I use “int argc, char \*argv[]” to replace “scanf”. At first I didn’t know how it would work. But later I start to know that I can make use of the already stored input by using “atof”. I’ve known the use of “define”, and also, I’ve rememberd that (4.0f/3.0f)==1.75 while (4/3)==1.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Code:

#include<stdio.h>

#define PI 3.14

int main(int argc, char \*argv[]){

float r,v;

r=atof(argv[1]);

v= (4.0f/3.0f) \* PI \* r \* r \* r;

printf("%.2f\n", v);

return 0;

}

Compilation:

gcc -o hw1\_1 hw1\_1.c

Execution:

./hw1\_1 3.0

Output:

113.04

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Report: hw1\_2

Author: F74082086 王維瀚 <leowang3268@gmail.com>

Class: 乙班

Description:

The problem is similar to the first one, so I writed the code according to the concept. But there’s a little problem. I found that I could only use atof instead of atoi to let the program run, which was weird since I have already declared x to be a float. Moreover, the answer would print the right float no matter I typed x as an integer or a float. Maybe I would go deeper to find the exact meaning of “atoi”.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Code:

#include <stdio.h>

int main(int argc, char \*argv[]){

float x, ans;

x=atoi(argv[1]);

ans=3\*x\*x\*x\*x\*x+2\*x\*x\*x\*x-5\*x\*x\*x-x\*x+7\*x-6;

printf("%.2f\n",ans);

return 0;

}

Compilation:

gcc -o hw1\_2 hw1\_2.c

Execution:

./hw1\_2 2.0

Output:

92.00

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Report: hw1\_3

Author: F74082086 王維瀚 <leowang3268@gmail.com>

Class: 乙班

Description:

I tried to figure out how I could apply “Hirner’s Rule” to the code, so I used something like this:

for(int i=n-1; i >= 0 ; --i){

ans= ans\* x + a[i];}

But there was something wrong, So finally I could just use the formula from the problem.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Code:

#include <stdio.h>

int main(int argc, char \*argv[]){

float x, ans;

x=atoi(argv[1]);

ans=((((3\*x+2)\*x-5)\*x-1)\*x+7)\*x-6;

printf("%.2f\n",ans);

return 0;

}

Compilation:

gcc -o hw1\_3 hw1\_3.c

Execution:

./hw1\_3 2.0

Output:

92.00