

Εργαστήριο Εισαγωγή στον Προγραμματισμό

Εργαστήριο 03

Δομές επανάληψης

Βασιλόπουλος Διονύσης

Ε.ΔΙ.Π. Τμήματος Πληροφορικής & Τηλεπικοινωνιών

4^ο Εργαστήριο

Άθροισμα αριθμών 1 .. 100

```
#include <stdio.h>

int sum() {
    int i=0;
    int temp=0;
    while (i<100) {
        temp+=++i;
    }
    return temp;
}

int main()
{
    printf("The sum is: %d \n", sum());
    return 0;
}
```

Όταν χρησιμοποιείτε μεταβλητές για υπολογισμούς σχεδόν πάντα χρειάζεται να την αρχικοποιείτε (συνήθως στο 0).

4^ο Εργαστήριο

Άθροισμα αριθμών 1, 1/4, 1/9 .. 1/10000

```
#include <stdio.h>

float basel() {
//int i=0;
float temp=0;
for (int i=1;i<=100;i++) {
    //temp+=(float)1/(i*i); //Solution 1
    temp+=1.0/(i*i); //Solution 2
    printf("temp: %f, i: %d\n", temp, i);
}
return temp;
}

int main()
{
printf("The sum is: %f \n", basel());
return 0;
}
```

`temp+=1/(i*i);`

Σε αυτή την περίπτωση το temp είναι πάντα 0 εκτός της 1^{ης} επανάληψης. Οπότε το άθροισμα γίνεται 1.

Λάθος και το:

`temp+=(float)(1/(i*i));`

https://www.tutorialspoint.com/cprogramming/c_type_casting.htm

4^ο Εργαστήριο

Άθροισμα αριθμών 1, 1/4, 1/9 .. 1/10000,

```
#include <stdio.h>

float basel() {
//int i=0;
float temp=0;
for (int i=1;i<=100;i++) {
    //temp+=(float)1/(i*i);
    temp+=1.0/(i*i);
    printf("temp: %f, i: %d\n", temp, i);
}
return temp;
}

int main()
{
printf("The sum is: %f \n", basel());
return 0;
}
```

```
#include <stdio.h>

float basel(int times) {
float temp=0;
for (int i=1;i<=times;i++) {
    temp+=1.0/(i*i);
    printf("temp: %f, i: %d\n", temp, i);
}
return temp;
}

int main()
{
printf("The sum is: %f \n", basel(200));
return 0;
}
```

Basel problem
->1.644934

4^ο Εργαστήριο

Pi (<https://www.imdb.com/title/tt0138704/>)

```
#include <stdio.h>
#include <math.h>

float pi_approx() {
    int i=0;
    float temp=0;
    do {
        i++;
        temp+=1.0/(i*i);
        printf("temp: %f, i: %d\n", temp, i);
    } while (i<100);
    return sqrt(6*temp);
}

int main()
{
    printf("The sum is: %f \n", pi_approx());
    return 0;
}
```

4^ο Εργαστήριο

Pi (with precision) - float

```
#include <stdio.h>
#include <math.h>

float pi_approx_detail() {
    int i=0;
    float temp=0;
    float new_term;

    do {
        i++;
        //new_term=1.0/(i*i);
        new_term=1.0/((float)i*(float)i);
        temp+=new_term;
        printf("temp: %f, i: %d, term: %.20f \n", temp, i, new_term);
    } while (new_term>=1e-15);
    return sqrt(6*temp);
}

int main()
{
    printf("The sum is: %f \n", pi_approx_detail());
    return 0;
}
```

Stops at i=46341

Term=-0.000000000046566228651

Sum= 3.141393

Stops at i=31622778

Term=0.0000000000000000100000

Sum= 3.141393

4^ο Εργαστήριο

Pi (with precision) - double

```
#include <stdio.h>
#include <math.h>

float pi_approx_detail() {
    int i=0;
    double temp=0;
    double new_term;

    do {
        i++;
        //new_term=1.0/(i*i);
        new_term=1.0/((double)i*(double)i);
        temp+=new_term;
        printf("temp: %f, i: %d, term: %.20f \n", temp, i, new_term);
    } while (new_term>=1e-15);
    return sqrt(6*temp);
}

int main()
{
    printf("The sum is: %f \n", pi_approx_detail());
    return 0;
}
```

Stops at i=46341
Term= -0.00000000046566229193
Sum= 3.141572

Stops at i=31622777
Term=0.000000000000000100000
Sum= 3.141593

<https://codefinity.com/blog/Float-vs-Double>

4^ο Εργαστήριο

srand – rand functions

https://www.tutorialspoint.com/c_standard_library/c_function_rand.htm

https://www.tutorialspoint.com/c_standard_library/c_function_srand.htm

srand:

Reproducibility: If you use the same seed, you'll get the same sequence of numbers. This can be useful for debugging or when you need reproducible results.

Single Call: Generally, srand() is only called once in a program. Calling srand() multiple times with the same seed will reset the sequence each time, which might not be desirable.

4^ο Εργαστήριο

Libraries

https://www.tutorialspoint.com/c_standard_library/stdio_h.htm

https://www.tutorialspoint.com/c_standard_library/stdlib_h.htm

https://www.tutorialspoint.com/c_standard_library/math_h.htm