

# Προγραμματισμός Ι

## Εργαστήριο 6

Διδάσκων: Χρήστος Δίου

Βασισμένο στο υλικό του κ. Δημήτρη Μιχαήλ

### 1 Ελάχιστος

Γράψτε μια συνάρτηση

```
1 int min2(int a, int b);
```

που να επιστρέφει τον ελάχιστο 2 ακεραίων. Στην συνέχεια γράψτε την συνάρτηση

```
1 int min3(int a, int b, int c);
```

που να επιστρέφει τον ελάχιστο 3 ακεραίων. Χρησιμοποιήστε την `min2` για να είναι πιο εύκολη η υλοποίηση. Τέλος γράψτε και τις

```
1 int min4(int a, int b, int c, int d);  
2 int min5(int a, int b, int c, int d, int e);
```

που επιστρέφουν τον ελάχιστο 4 και 5 ακεραίων αντίστοιχα. Χρησιμοποιήστε τις `min2` και `min3` για πιο εύκολη υλοποίηση.

Για να ελέγξετε την ορθότητα γράψτε ένα πρόγραμμα που να ζητάει 5 ακεραίους από τον χρήστη και να τυπώνει στην έξοδο τον ελάχιστο αυτών.

### 2 Προσέγγιση της $e$

Η τιμή της μαθηματικής σταθεράς  $e$  είναι 2.71828182845..., που ισοδυναμεί με την άπειρη σειρά:

$$e = \sum_{k=0}^{\infty} \frac{1}{k!}$$

Γράψτε ένα πρόγραμμα που να ζητάει από τον χρήστη μία θετική ακέραια τιμή  $1 \leq n \leq 10$  και μετά να προσεγγίζει την τιμή του  $e$  με τους πρώτους  $n$  όρους της παραπάνω άπειρης σειράς. Το πρόγραμμα σας πρέπει να τυπώνει αυτή την προσέγγιση.

Παράδειγμα εκτέλεσης του προγράμματος σας είναι:

Give number of terms:

10

With 1 terms  $e = 1.00000000000000000000$ With 2 terms  $e = 2.00000000000000000000$ With 3 terms  $e = 2.50000000000000000000$ With 4 terms  $e = 2.66666666666666666666$ With 5 terms  $e = 2.70833333333333333333$ With 6 terms  $e = 2.71666666666666666666$ With 7 terms  $e = 2.71805555555555555555$ With 8 terms  $e = 2.71825396825396836675$ With 9 terms  $e = 2.71827876984127003723$ With 10 terms  $e = 2.71828152557319224769$

### 3 Δυνάμεις

Γράψτε μια συνάρτηση που να δέχεται δύο ακέραιους αριθμούς  $x$  και  $n \geq 0$  και να υπολογίζει την δύναμη  $x^n$ . Η συνάρτηση θα πρέπει να έχει το αρχέτυπο:

```
1 long power(int x, int n);
```

και να χρησιμοποιεί τον τύπο  $x^n = x \cdot x \dots x$ . Χρησιμοποιήστε επανάληψη. Στη περίπτωση όπου  $x = 0$  και  $n = 0$  θεωρούμε<sup>1</sup> πως  $0^0 = 1$ .

Χρησιμοποιήστε το παρακάτω πρόγραμμα για να ελέγξετε την ορθότητα της συνάρτησής σας.

```
1 int main() {  
2     int i, j;  
3  
4     for(i=0; i<6; i++) {  
5         for(j=0; j<8; j++) {  
6             printf("%6ld ", power(i, j));  
7         }  
8         printf("\n");  
9     }  
10    return 0;  
11 }
```

Μετά την εκτέλεση του θα πρέπει να δείτε στην έξοδο:

1	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1
1	2	4	8	16	32	64	128
1	3	9	27	81	243	729	2187
1	4	16	64	256	1024	4096	16384
1	5	25	125	625	3125	15625	78125

---

<sup>1</sup><http://mathforum.org/dr.math/faq/faq.0.to.0.power.html>