

Άσκηση 1

A) Γράψτε σε κώδικα τη συνάρτηση

$$f(x) = x^4 + 4x^3 - 2x - 1$$

B) Υπολογίστε τις διάφορες τιμές της συνάρτησης $f(x)$ χρησιμοποιώντας τη συνάρτηση που φτιάξατε στο (A), για όλες τις ακέραιες τιμές του x εντός του διαστήματος $[-5, 2]$. Προφανώς αυτό πρέπει να γίνει με μία επαναληπτική διαδικασία (for, while,...). Ο κώδικάς σας πρέπει επίσης να αποθηκεύει σε μία μεταβλητή την ελάχιστη τιμή της συνάρτησης (min_y), για τα x που ελέγξατε, καθώς και το συγκεκριμένο x το οποίο οδήγησε σε αυτήν την ελάχιστη τιμή (x_for_min_y).

Παράδειγμα δομής:

```
def my_function(x_value):
```

```
    result = ...
```

```
    return result
```

```
y_test = my_function(4)
```

```
min_y = 100000
```

```
x_for_min_y = 100000
```

```
for ...:
```

```
    result = my_function(...)
```

```
    #γράψτε κώδικα για να αποθηκεύστε τα min_y και x_for_min_y
```

Γ) Τροποποιείστε τον Κώδικα του ερωτήματος (B), ώστε να ελέγχετε όλες τις τιμές στο διάστημα $[-5, 2]$, αλλά με βήμα 0.01, δηλαδή $[-5.00, -4.99, -4.98, \dots, 1.98, 1.99, 2.00]$. Ίσως σας διευκολύνει να δουλέψετε με while. Hint: Μπορείτε να δουλέψετε και με for στο διάστημα $[-500, 200]$ αλλά προφανώς πρέπει να προσαρμόσετε τον κώδικα εντός του for κατάλληλα.

Άσκηση 2

Γράψτε μία συνάρτηση που θα τυπώνει ένα δέντρο. Η συνάρτηση πρέπει να παίρνει ένα όρισμα το οποίο θα αντιστοιχεί στο ύψος του δέντρου.

Για ύψος 1, το αποτέλεσμα θέλουμε να είναι:

```
**
```

Για ύψος 2, το αποτέλεσμα θέλουμε να είναι:

```
  **  
***
```

Για ύψος 3, το αποτέλεσμα θέλουμε να είναι:

```
  **  
***  
****
```

Παράδειγμα δομής:

```
def print_tree(height):
```

```
...
```

```
print_tree(1)
```