

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 2.1

Πρόβλημα

- Να γίνει πρόγραμμα C, το οποίο θα διαβάζει δύο ακέραιους αριθμούς x , n και θα υπολογίζει την τιμή της σειράς :

$$\tilde{e}^x = \sum_{i=0}^n \frac{x^i}{i!} = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

Για τον υπολογισμό των παραγοντικών να χρησιμοποιηθεί συνάρτηση – function (μέθοδος).

Αλγόριθμος `main()`

1. **Διαβάζουμε** την τιμή του x και του n (ο αριθμός των όρων που θα χρησιμοποιηθούν).
2. **Δίνουμε** την **αρχική** τιμή 1 στη μεταβλητή `myexp`.
3. **Για τον κάθε** όρο $i = 1..n$
 - a) Υπολογίζουμε τον κάθε όρο με τη χρήση της συνάρτησης `pow()` για τον αριθμητή και της συνάρτησης `iparag()` για τον παρονομαστή.
 - b) **Προσθέτουμε** τον νέο όρο στο `myexp`.
4. **Εμφανίζουμε** την τιμή του x , του `myexp` και του `exp(x)` .

Αλγόριθμος Υπολογισμού `i!` – function `parag()`

1. **Δίνουμε** αρχική τιμή το 1 στο γινόμενο `iparag` (`iparag ← 1`)
2. **Για** τις τιμές του μετρητή j από το 1 μέχρι και το i
Πολλαπλασιάζουμε το j με το `iparag` (`iparag ← iparag * j`)
3. **Επιστρέφουμε** την τιμή του γινομένου `iparag`

2.1 Συναρτήσεις (Functions)

- Εκτός απ' τη δυνατότητα να γράφουμε συναρτήσεις μιας γραμμής με τη macro-εντολή `#define`, η C μας παρέχει και τη δυνατότητα να γράφουμε συναρτήσεις που αποτελούνται από πολλές εντολές, αντίστοιχες με τις μεθόδους της Java. Η γενική τους μορφή είναι :

<Τύπος_Function > <Όνομα_Function> (Τυπικές_Παράμετροι)

Δηλώσεις του τύπου των Παραμέτρων

```
{  
    εντολές  
}
```

Παρατηρήσεις

- ◆ Η Συνάρτηση πρέπει να τοποθετείται **στην αρχή ή στο τέλος** του προγράμματος. Αν τοποθετηθεί **μετά** το πρόγραμμα main(), τότε ο τύπος της συνάρτησης πρέπει να δηλωθεί και στο κυρίως πρόγραμμα.
- ◆ Μπορεί να μην υπάρχει καμία Τυπική Παράμετρος.
- ◆ Οι παρενθέσεις στον ορισμό της Συνάρτησης είναι υποχρεωτικές.
- ◆ Το όνομα της συνάρτησης πρέπει να ακολουθεί τους κανόνες των μεταβλητών.
- ◆ Η συνάρτηση πρέπει να επιστρέφει μία τιμή. Η τιμή επιστρέφει με την εντολή **return** <μεταβλητή> , όπου <μεταβλητή> = τοπική μεταβλητή της συνάρτησης.

2.2 Χρήση Συναρτήσεων

- Μια Συνάρτηση μπορεί να χρησιμοποιηθεί σαν παράμετρος της εντολής printf, σε κάποια συνθήκη ή στο δεύτερο μέλος οποιασδήποτε εντολής ανάθεσης, όπως και κάθε μεταβλητή. Η γενική της μορφή είναι :

var = <Όνομα_function> (Πραγματικές_Παράμετροι)

όπου :

var, Πραγματικές_Παράμετροι = Μεταβλητές

Παρατηρήσεις

- ◆ Οι Πραγματικές Παράμετροι μπορεί να είναι Σταθερές, Μεταβλητές, Ονόματα Πινάκων, Στοιχεία Πινάκων, Αριθμητικές Εκφράσεις. Άλλες Συναρτήσεις.
- ◆ Η συνάρτηση σε κάθε κλήση της δίνει **μια** τιμή.
- ◆ Ο αριθμός, ο τύπος και η σειρά των Πραγματικών Παραμέτρων στη χρήση της συνάρτησης στο κυρίως πρόγραμμα πρέπει να συμφωνεί με τον αριθμό, τον τύπο και τη σειρά των Τυπικών Παραμέτρων στον Ορισμό της συνάρτησης.
- ◆ Τα ονόματα των Πραγματικών Παραμέτρων μπορεί να είναι ή να μην είναι ίδια με τα ονόματα των Τυπικών Παραμέτρων.
- ◆ Η εντολή **return** μπορεί να εμφανίζεται πολλές φορές στη Συνάρτηση.

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 2.2

Πρόβλημα

- Να γίνει πρόγραμμα C, το οποίο θα διαβάζει δύο ακέραιους αριθμούς x , n και θα υπολογίζει την τιμή της σειράς :

$$\tilde{e}^x = \sum_{i=0}^n \frac{x^i}{i!} = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

Ο κάθε όρος θα υπολογίζεται από τον προηγούμενο με τη χρήση αναγωγικού τύπου.

Αλγόριθμος `main()`

1. **Διαβάζουμε** την τιμή του x και του n (ο αριθμός των όρων που θα χρησιμοποιηθούν).
2. **Δίνουμε** την **αρχική** τιμή 1 στη μεταβλητή `i`.
3. **Δίνουμε** την **αρχική** τιμή 1 στη μεταβλητή `oros`.
4. **Δίνουμε** την **αρχική** τιμή 1 στη μεταβλητή `myexp`.
5. **Για τον κάθε** όρο $i = 1..n$
 - a) Υπολογίζουμε τον κάθε όρο από τον προηγούμενο με τη χρήση του αναγωγικού τύπου `oros ← oros *x/i`
 - b) **Προσθέτουμε** τον νέο όρο στο `myexp`.
6. **Εμφανίζουμε** την τιμή του x , του `myexp` και του `exp(x)`.

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 2.3

Πρόβλημα

- Να γίνει πρόγραμμα C, το οποίο θα διαβάζει έναν ακέραιο αριθμό x και θα υπολογίζει την τιμή της σειράς :

$$\tilde{e}^x = \sum_{i=0}^n \frac{x^i}{i!} = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

Ο κάθε όρος θα υπολογίζεται από τον προηγούμενο με τη χρήση αναγωγικού τύπου. Θα προστίθενται όροι, για όσο η τιμή τους ξεπερνάει το 10^{-15} .

Αλγόριθμος `main()`

1. Διαβάζουμε την τιμή του x .
2. Δίνουμε την αρχική τιμή 1 στη μεταβλητή `i`.
3. Δίνουμε την αρχική τιμή 1 στη μεταβλητή `oros`.
4. Δίνουμε την αρχική τιμή 1 στη μεταβλητή `myexp`.
5. Για όσο `oros > 10-15`
 - a) Υπολογίζουμε τον κάθε όρο από τον προηγούμενο με τη χρήση του αναγωγικού τύπου `oros ← oros * x / i`
 - b) Προσθέτουμε τον νέο όρο στο `myexp`.
6. Εμφανίζουμε την τιμή του x , του `myexp` και του `exp(x)`.

ΕΡΓΑΣΤΗΡΙΑΚΗ ΑΣΚΗΣΗ 2.4

Πρόβλημα

- Να γίνει πρόγραμμα C, το οποίο θα διαβάζει έναν ακέραιο αριθμό x και θα υπολογίζει την τιμή της σειράς :

$$\tilde{e}^x = \sum_{i=0}^n \frac{x^i}{i!} = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!}$$

Ο κάθε όρος θα υπολογίζεται από τον προηγούμενο με τη χρήση αναγωγικού τύπου. Θα προστίθενται όροι, για όσο η τιμή τους ξεπερνάει το 10^{-15} και ο υπολογισμός του αθροίσματος θα γίνεται με την κλήση της συνάρτησης **myexp(x)**.

Αλγόριθμος main ()

1. Διαβάζουμε την τιμή του x .
2. Εμφανίζουμε την τιμή του x , του **myexp (x)** και του **exp (x)** .

Αλγόριθμος function myexp(x)

1. Δίνουμε την αρχική τιμή 1 στη μεταβλητή **i**.
2. Δίνουμε την αρχική τιμή 1 στη μεταβλητή **oros**.
3. Δίνουμε την αρχική τιμή 1 στη μεταβλητή **sum**.
4. Για όσο **oros** > 10^{-15}
 - a) Υπολογίζουμε τον κάθε όρο από τον προηγούμενο με τη χρήση του αναγωγικού τύπου **oros** ← **oros** * x/i
 - b) Προσθέτουμε τον νέο όρο στο **sum**.
5. Επιστρέφουμε την τιμή του **sum**.