2ο Σφάλματα

2.1 Εισαγωγικά

Από όσα αναφέραμε μέχρι τώρα φαίνεται ότι, όταν μιλάμε για αριθμητική λύση ενός προβλήματος, εννοούμε μια προσέγγιση στη λύση. Δηλαδή υπάρχει πάντοτε ένα σφάλμα. Το μεγαλύτερο μέρος της αριθμητικής ανάλυσης ασχολείται με τα σφάλματα και πώς αυτά μπορούν να ελαχιστοποιηθούν.

Υπάρχουν πολλά είδη σφαλμάτων, και πρώτα απ' όλα:

1. Το σφάλμα στρογγύλευσης κατά την αποθήκευση των αριθμών στον υπολογιστή.
2. Η αριθμητική μέθοδος που επιλέγεται για να λυθεί αριθμητικά ένα πρόβλημα είναι συσχετισμένη με το δικό της σφάλμα.
3. Τα σφάλματα που αναπτύσσονται κατά την πορεία των αριθμητικών υπολογισμών.
4. Τα σφάλματα που οφείλονται στα πεπερασμένα όρια μεγέθους των αριθμών που μπορεί να χειρισθεί ένας υπολογιστής.

Στο κεφάλαιο αυτό θα περιγραφούν διάφορα είδη σφαλμάτων, πηγές σφαλμάτων και διάφορα παραδείγματα για την καλύτερη κατανόησή τους.

Καλούμε γενικά σφάλμα τη διαφορά

$$(\boldsymbol{\sigma\varphi}ά\boldsymbol{\lambda\mu\alpha}) = (\boldsymbol{\alpha\lambda\eta\theta}ή\boldsymbol{\varsigma}\ \boldsymbol{\tau\iota\mu}ή) - (\boldsymbol{\pi\rho o\sigma\varepsilon\gamma\gamma\iota\sigma\tau\iota\kappa}ή\ \boldsymbol{\tau\iota\mu}ή)$$

Επίσης καλούμε σχετικό σφάλμα το πηλίκο

$$(\sigma\chi\tau\iota\kappa ό\ \sigma\varphi ά\lambda\mu\alpha) = \frac{(\sigma\varphi ά\lambda\mu\alpha)}{(\alpha\lambda\eta\theta ή\varsigma\ \tau\iota\mu ή)}$$

Δηλαδή το σχετικό σφάλμα είναι ένα μέτρο του σφάλματος σε σχέση με το μέγεθος της αληθούς τιμής.

Όταν ένας αριθμός x αποθηκεύεται σε έναν υπολογιστή, τότε θα μιλάμε για έναν άλλο αριθμό, έστω x*, που είναι μια προσέγγιση του x. Θα καλούμε σφάλμα E ή σφάλμα στρογγύλευσης (round off error) τη διαφορά

$E = x - x*$ ή $E = |x - x*|$ απόλυτο σφάλμα

και σχετικό σφάλμα

$$a = \frac{x - x^*}{x}$$

Παράδειγμα

Αν ο αριθμός π=3.14159265..... προσεγγισθεί με τον αριθμό 22/7=3.1428571...., τότε το σφάλμα Ε και το σχετικό σφάλμα ε θα είναι αντίστοιχα

Ε=π-(22/7)=-0.00126

ε=(-0.00126)/π=-0.000402

2.2 Μετάδοση σφαλμάτων

Στην παράγραφο αυτή θα ασχοληθούμε τον τρόπο με τον οποίο τα σφάλματα στρογγύλευσης κατά την αποθήκευση των αριθμών σε έναν υπολογιστή μεταδίδονται στη διαδικασία εκτέλεσης των τεσσάρων αριθμητικών πράξεων.

Πριν όμως δούμε πως μεταδίδονται τα σφάλματα θα πρέπει να βρούμε μια σχέση της αληθούς με την προσεγγιστική τιμή.

Έστω x ένας πραγματικός αριθμός, ο οποίος στην εκθετική έκφραση είναι

$$x=\sigma\alpha 2^{e}$$

Όπου σ είναι το πρόσημο, α η mantissa και e ο εκθέτης.

Ο αριθμός x αποθηκεύεται σε έναν υπολογιστή. Αν ο υπολογιστής, για παράδειγμα, διαθέτει 3 χαρακτήρες (bytes) για την αποθήκευση της mantissa, τότε ν=3x8=24 δηλαδή θα απαιτηθούν 24 δυαδικές θέσεις. Συνεπώς το σφάλμα στρογγύλευσης δε μπορεί να υπερβαίνει την τιμή $E=2^{-24}$. Έτσι ο α θα προσεγγισθεί από τον αριθμό α* και θα ισχύει προφανώς

α*=α+Ε όπου Ε το σφάλμα.

Δηλαδή

$x^*=\sigma\alpha^* 2^{e}=\sigma(\alpha+E)2^{e}=\sigma\alpha 2^{e}(1+E/\alpha)=x(1+\varepsilon)$

όπου $\varepsilon=E/\alpha$ το σχετικό σφάλμα.

Επειδή $0.5 \prec \alpha \leq 1$ τότε το $\varepsilon \prec 2^{-24}$.

Θεωρούμε τώρα δυο αριθμούς x και y οι οποίοι όταν αποθηκεύονται σε έναν υπολογιστή θα ισχύει

$x^*=x(1+\varepsilon_1)$ και $y^*=y(1+\varepsilon_2)$

όπου ε_1 και ε_2 τα αντίστοιχα σχετικά σφάλματα. Θα δούμε στη συνέχεια πως τα σφάλματα μεταδίδονται κατά την εκτέλεση των τεσσάρων αριθμητικών πράξεων.

Πολλαπλασιασμός και διαίρεση

Έστω $z=xy$ το ακριβές γινόμενο των x και y και $z^*=x^*y^*$ το αντίστοιχο γινόμενο των αποθηκευμένων αριθμών. Τότε

$z^*=x^*y^*=x(1+\varepsilon_1)y(1+\varepsilon_2)=xy(1+\varepsilon_1+\varepsilon_2+\varepsilon_1\varepsilon_2) \cong z(1+\varepsilon_1+\varepsilon_2) = z(1+\varepsilon)$

όπου ο όρος $\varepsilon_1\varepsilon_2$ παραλείπεται ως αμελητέα ποσότητα σε σχέση με τα ε_1 και ε_2 και $\varepsilon=\varepsilon_1+\varepsilon_2$.

Συνεπώς τη μεγαλύτερη τιμή που μπορεί να πάρει το ε είναι το άθροισμα των απόλυτων τιμών των ε_1 και ε_2.

Στο ίδιο συμπέρασμα καταλήγουμε και με την πράξη της διαίρεσης.

$z^*=x^*/y^*=x(1+\varepsilon_1)/y(1+\varepsilon_2)=x/y(1+\varepsilon_1)(1+\varepsilon_2)^{-1}=z(1+\varepsilon_1)(1-\varepsilon_2+\varepsilon_2^2$

$-\varepsilon_2^3+ . \quad . .)=z(1+\varepsilon_1-\varepsilon_2)=z(1+\varepsilon)$

Θεωρώντας επίσης αμελητέες όλες τις άλλες ποσότητες εκτός των ε_1 και ε_2.

Πρόσθεση και αφαίρεση

Αν $z=x+y$, τότε το άθροισμα των αντίστοιχων αποθηκευμένων x^* και y^* θα είναι

$$\begin{aligned} z^* = x^* + y^* &= x(1+\varepsilon_1) + y(1+\varepsilon_2) = (x+y) + (x\varepsilon_1 + y\varepsilon_2) \\ &= (x+y)\left(1 + \frac{x\varepsilon_1 + y\varepsilon_2}{x+y}\right) = z(1+\varepsilon) \end{aligned}$$

όπου

$$\varepsilon = \frac{x\varepsilon_1 + y\varepsilon_2}{x + y}$$

Από τις παραπάνω σχέσεις συμπεραίνουμε ότι όταν προστίθενται δυο ετερόσημοι αριθμοί x και y υπάρχει περίπτωση να έχουμε μεγάλο σφάλμα και ιδιαίτερα στην περίπτωση που οι x και y είναι πολύ κοντά ο ένας στον άλλο. Το σφάλμα αυτό καλούμε συνήθως σφάλμα ***απώλειας σημαντικών ψηφίων*** και θα το εξετάσουμε στην επόμενη παράγραφο.

Αν έχουμε να κάνουμε μεγάλο αριθμό προσθέσεων, τότε τα σφάλματα στρογγύλευσης συσσωρεύονται στο τελικό αποτέλεσμα. Το πρόβλημα είναι πως μπορούμε να ελαχιστοποιήσουμε τέτοια σφάλματα.

Έστω, για παράδειγμα, ότι έχουμε να υπολογίσουμε το άθροισμα

$\Sigma=\alpha_1+\alpha_2+\alpha_3+ \ . \ . \ . +\alpha_\nu$

και α_1^*, α_2^*, . . . , οι αντίστοιχοι αποθηκευμένοι αριθμοί, τότε

$\Sigma_2=\alpha_1^*+\alpha_2^*=(\alpha_1+\alpha_2)(1+\varepsilon_2)$

$\Sigma_3=\Sigma_2+\alpha_3^* =(\Sigma_2+\alpha_3)(1+\varepsilon_3)$

$\Sigma_4=\Sigma_3+\alpha_4^* =(\Sigma_3+\alpha_4)(1+\varepsilon_4)$

.

.

$\Sigma_\nu=\Sigma_{\nu-1}+\alpha_\nu^*=(\Sigma_{\nu-1}+\alpha_\nu)(1+\varepsilon_\nu)$

όπου Σ_ν είναι η προσέγγιση του Σ.

Το ολικό συσσωρευμένο σφάλμα

$\Sigma-\Sigma_\nu = -\alpha_1(\varepsilon_2+\varepsilon_3+\ldots+\varepsilon_\nu) - \alpha_2 (\varepsilon_2+\varepsilon_3+\ldots\varepsilon_\nu) - \alpha_3(\varepsilon_3+\varepsilon_4+\ldots+\varepsilon_\nu) -$

$\alpha_4(\varepsilon_4+\varepsilon_5+\ldots+\varepsilon_\nu) - \ . \ . \ . \ - \alpha_{\nu-1} (\varepsilon_{\nu-1}+\varepsilon_\nu) - \alpha_\nu\varepsilon_\nu$

Βλέποντας προσεκτικά τις παραπάνω σχέσεις παρατηρούμε ότι οι α1 και α2 πολλαπλασιάζονται με το ολικό άθροισμα όλων των σφαλμάτων, ενώ όσο αυξάνει ο δείκτης του α οι όροι του αθροίσματος των σφαλμάτων μειώνονται κατά έναν όρο και ο τελευταίος όρος α_ν πολλαπλασιάζεται μόνο με το σφάλμα ε_ν.

Για να μπορέσουμε λοιπόν να ελαχιστοποιήσουμε το σφάλμα θα πρέπει ο όρος που περιλαμβάνει το άθροισμα όλων των σφαλμάτων να πολλαπλασιάζεται με την απολύτως μικρότερη τιμή του a_j. Αυτό επιτυγχάνεται αν διατάξουμε τα a_j, έτσι ώστε:

$$\alpha 1 \prec \alpha 2 \prec \alpha 3 \prec \alpha 4 \prec \ldots \prec \alpha\nu - 1 \prec \alpha\nu$$

Παράδειγμα

Θεωρούμε το άθροισμα $s=(a_1+a_2+a_3+\ldots+a_{1000})$ όπου $a_j=1/j$

Αν εργασθούμε με έναν υπολογιστή με επτά σημαντικά ψηφία ακρίβεια και διατάσσοντας τα a_j πρώτα κατά αύξουσα σειρά και μετά κατά φθίνουσα παρατηρούμε ότι τα αντίστοιχα σφάλματα επιβεβαιώνουν τα όσα είπαμε παραπάνω.

Η ακριβής τιμή s=7.48547087. Κατ' αρχάς υπολογίζουμε το άθροισμα έτσι ώστε τα a_j να αυξάνονται διαδοχικά και έχουμε το αποτέλεσμα s=7.485472, ενώ αν τα a_j ελαττώνονται, τότε έχουμε το αποτέλεσμα s=7.485479.

2.3 Απώλεια σημαντικών ψηφίων

Για να γίνει πιο κατανοητή η έννοια της απώλειας σημαντικών ψηφίων θα πρέπει να τη δούμε μέσα από συγκεκριμένα αριθμητικά παραδείγματα.

Έστω x* και y* δυο αριθμοί κινητής υποδιαστολής μήκους επτά σημαντικών ψηφίων

$x^*=(0.3645521)10^1$ και $y^*=(0.3645467)10^1$

που προέκυψαν από την αποθήκευση των x και y μέσω μιας γλώσσας προγραμματισμού που χρησιμοποιεί τρεις χαρακτήρες (bytes) για τη mantissa. Παρατηρούμε ότι οι δυο αριθμοί αυτοί έχουν τα τέσσερα πρώτα σημαντικά ψηφία τα ίδια.

Αν αφαιρέσουμε τώρα τους αριθμούς x* και y* , δηλαδή υπολογίσουμε τη διαφορά z*-x*-y* θα έχουμε

3.645521
3.645467
----
z*=0.000054

Για την ίδια πράξη ο υπολογιστής θα μας δώσει διαφορετικό αποτέλεσμα (γιατί;) z*=0.538826 10^{-4}. Είναι προφανές ότι το δεύτερο σημαντικό ψηφίο (δηλαδή το 3 στην περίπτωση που την πράξη έκανε ο υπολογιστής ή το 4 στην περίπτωση της αφαίρεσης με το κλασικό τρότρόπο) προέκυψε από την αφαίρεση των έβδομων σημαντικών ψηφίων των x* και y*, τα οποία πιθανόν να μη συμφωνούν με τα αντίστοιχα έβδομα ψηφία των x και y αφού έγινε στρογγύλευση. Έτσι το z* θα συμφωνεί με το z μόνο στο πρώτο σημαντικό ψηφίο (δηλαδή στο 5) ενώ τα x* και y* συμφωνούν με τα αντίστοιχά τους x και y στα πρώτα έξη σημαντικά ψηφία. Δηλαδή κατά την αφαίρεση των x και y έχουμε απώλεια πέντε σημαντικών ψηφίων. Τα σφάλματα που προκύπτουν από την απώλεια σημαντικών ψηφίων είναι αρκετά σημαντικά και τα αποτελέσματα μπορεί να είναι εντελώς αναξιόπιστα. Το παρακάτω παράδειγμα δίνει μια πραγματική εικόνα του φαινομένου αυτού.

Παράδειγμα

Έστω ότι θέλουμε να βρούμε την αριθμητική τιμή της συνάρτησης

$$f(x) = \frac{x^2 - 2(1 - \cos x)}{2x^2(1 - \cos x)}$$

Αν το x πάρει την τιμή 0.001 τότε ο υπολογιστής δίνει f(0.001)= 48576.11, ενώ η ακριβής τιμή είναι f(0.001)=0.08333333. Για x=3.0 έχουμε f(3.0)=0.1401461 και δεν παρατηρείται καμιά σοβαρή απώλεια στην ακρίβεια.

Σύμφωνα με αυτά που είπαμε παραπάνω το λάθος θα πρέπει να οφείλεται σε σφάλμα απώλειας σημαντικών ψηφίων και συγκεκριμένα στον όρο 1 - cosx, επειδή για μικρά x η τιμή του cosx είναι πολύ κοντά στο 0 με αποτέλεσμα να αφαιρούνται αριθμοί που είναι απολύτων πολύ κοντά ο ένας στον άλλο και να έχουμε απώλεια σημαντικών ψηφίων κατά την πράξη της αφαίρεσης.. Η αποφυγή τέτοιων λαθών επιτυγχάνεται με την εξουδετέρωση της αιτίας που τα προκαλεί. Στη συγκεκριμένη περίπτωση η αποφυγή απώλειας σημαντικών ψηφίων επιτυγχάνεται αν αντικατασταθεί η συνάρτηση 1-cosx με μια άλλη που να μην έχει την πράξη της αφαίρεσης, δηλαδή

$$1 - \cos x = \frac{1 - \cos^2 x}{1 + \cos x} = \frac{\sin^2 x}{1 + \cos x}$$

Για πολύ μικρά x πιο κατάλληλη έκφραση είναι αυτή που προκύπτει αν αναπτύξουμε σε σειρά Taylor το οπότε μετά από στοιχειώδεις πράξεις έχουμε

$$f(x) = \frac{1}{12}(1 + \frac{x^2}{20} + \frac{x^4}{504} + \frac{x^6}{14400} + \cdots)$$

Αν αναπτύξουμε σχετικό πρόγραμμα υπολογίζεται ο παρακάτω πίνακας

x	(3.9)	(3.10)	(3.11)
5	.6579947	.6579946	.3812624
3	.1401461	.1401461	.138445
2	.1030707	.1030706	.1030159
1	.0876714	8.767116E-02	8.767113E-02
.5	8.438503E-0	8.438588E-02	8.438542E-02
.1	8.241541E-02	8.338165E-02	8.337501E-02.
05	6.714101E-02	8.334351E-02	8.334375E-02
.01	10.27182	8.496094E-02	8.333375E-02
.005	-54.24863	.0859375	8.333343E-02
.001	48576.11	0	8.333334E-02
.0005	194304.4	-.25	8.333334E-02

Η πρώτη στήλη του πίνακα περιέχει τις τιμές της μεταβλητής x και οι άλλες οι στήλες περιέχουν τις αντίστοιχες τιμές που δίνει το πρόγραμμα με τις τρεις διαφορετικές εκφράσεις της f(x). Για πολύ μεγάλα x οι δύο πρώτοι τύποι μας δίνουν πολύ καλά αποτελέσματα ενώ για μικρά x ο πιο κατάλληλος τύπος είναι ο τρίτος. Τα σφάλματα αυτά οφείλονται σε απώλεια σημαντικών ψηφίων.

2.4 Σφάλματα υπερχείλισης (overflow error)

Όπως αναφέραμε στο προηγούμενο κεφάλαιο, ο υπολογιστής δεν μπορεί να χειρισθεί αριθμούς που δεν περιέχονται σε ένα συγκεκριμένο διάστημα. Αναφέραμε επίσης πως, αν για την αποθήκευση του εκθέτη, χρησιμοποιείται ένας χαρακτήρας των 8 bits, τότε ο υπολογιστής μπορεί να χειρισθεί αριθμούς από $2.9x10^{-39}$ έως $1.7x10^{38}$. Αν στη διαδικασία εκτέλεσης αριθμητικών πράξεων προκύψει ένας αριθμός που δεν βρίσκεται στο διάστημα αυτό (π.χ. αν έχουμε διαίρεση με έναν πολύ μικρό αριθμό), τότε η γλώσσα BASIC δίνει ***μήνυμα σφάλματος υπερχείλισης*** και συγκεκριμένα ***overflow error.***

Σφάλματα υπερχείλισης μπορούν να αποφευχθούν αν είμαστε προσεκτικοί στη διαδικασία που εκτελούνται οι πράξεις. Για παράδειγμα, αν έχουμε να υπολογίσουμε την συνάρτηση e^x (την οποία συνήθως συμβολίζουμε exp(x)) με ανάπτυγμα σε σειρά Taylor, δηλαδή

$$\exp(x) = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots +$$

και γράψουμε ένα πρόγραμμα που να υπολογίζει κάθε όρο και να τον προσθέτει στον προηγούμενό του (δηλαδή υπολογίζονται τα x^r και το r! και βρίσκουμε το πηλίκο $x^r/r!$) και η διαδικασία τερματίζεται, όταν η τιμή του πηλίκου είναι απολύτως μικρότερη από 10^{-6} θα έχουμε.

Παρόλο που το αποτέλεσμα για x=9 είναι exp(x)=8103.083, έχουμε σφάλμα υπερχείλισης. Αυτό οφείλεται στο γεγονός πως σε μια ενδιάμεση αριθμητική πράξη χρειάσθηκε να υπολογισθεί ο 35!. Ο αριθμός αυτός είναι μεγαλύτερος από $1.7x10^{38}$ με αποτέλεσμα να μη βρίσκεται μέσα στα όρια που να μπορεί ο υπολογιστής να χειρίζεται αριθμούς και κατά συνέπεια να έχουμε σφάλμα υπερχείλισης.

Αν όμως κάνουμε ένα διαφορετικό πρόγραμμα, έτσι ώστε κάθε όρος να σχηματίζεται από τον προηγούμενο χωρίς να υπολογίζονται τα παραγοντικά των παρονομαστών, δηλαδή

$$(όρος)_{\nu+1} = (όρος)_{\nu} \frac{\chi}{\nu + 1}$$

τότε εκτός του γεγονότος ότι αποφεύγουμε το σφάλμα υπερχείλισης, κάνουμε και μεγάλη οικονομία στον αριθμό αριθμητικών πράξεων και συνεπώς οικονομία σε υπολογιστικό χρόνο. Γράψουμε ένα αντίστοιχο πρόγραμμα

και εκτελούμε το πρόγραμμα για διάφορα x θα έχουμε

Για x=1 exp(x)=1.718282
Για x=5 exp(x)=148.4132
Για x=9 exp(x)=8103.084
Για x=20 exp(x)=4.851652E+08

Ένας άλλος εναλλακτικός τρόπος για να αποφύγουμε το σφάλμα υπερχείλισης είναι η έκφραση

$$\exp(x) = 1 + \frac{x}{1}(1 + \frac{x}{2}\left(1 + \frac{x}{3}\left(1 + \frac{x}{4}\left(1 + \cdots \frac{x}{v-1}\left(1 + \frac{x}{v}\right)\right) \ldots \right)\right))$$

Στην περίπτωση αυτή θα πρέπει να προσδιορισθεί η τιμή του n. Από διάφορες δοκιμές για τη συγκεκριμένη ακρίβεια που θέλουμε προέκυψε η εξίσωση

$$v = int(3 * x + 12)$$

2.5 Ασκήσεις

1. Να βρεθούν οι ρίζες της εξίσωσης

$$x^2-23456x+7=0.$$

Σχόλια: Οι τιμές των ριζών της εξίσωσης με ακρίβεια 9 σημαντικών ψηφίων είναι

$$x_1=0.23455997\ 10^5$$

$$x_2=0.298431109\ 10^{-3}$$

Αν χρησιμοποιηθεί ο κλασικός τύπος εύρεσης των ριζών, δηλαδή

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

τότε η δεύτερη ρίζα θα βρεθεί $x_2=0$. Αν όμως χρησιμοποιηθεί ο τύπος

$$x_{1,2} = \frac{2c}{-b \pm \sqrt{b^2 - 4ac}}$$

τότε θα βρεθεί η σωστή απάντηση (γιατί;).

2. Να βρεθεί μια εναλλακτική έκφραση των παραστάσεων

1) $f(x) = \dfrac{x - \sin x}{\tan x}$

2) $g(x) = \sin(a + x) - \sin a$

3) $w(x) = \dfrac{e^x - 1}{x}$ $\text{για πολύ μικρά } \chi$

$\delta)\ r(x) = x\left(\sqrt{x+1} - \sqrt{x},\ \right)$ για μεγάλα x

έτσι ώστε να έχουμε τη δυνατόν μικρότερη απώλεια σημαντικών ψηφίων.

3. Να γραφεί ένα πρόγραμμα που να μετατρέπει πραγματικούς αριθμούς σε αριθμούς κινητής υποδιαστολής μήκους ν. Να χρησιμοποιηθεί το είδος στρογγύλευσης σαν περικοπή των ψηφίων.

4. Να γραφούν υπο-προγράμματα τα οποία να εκτελούν τις τέσσερις αριθμητικές πράξεις με μήκος αριθμών κινητής υποδιαστολής ν και με τιμή του ν μικρότερη από αυτή της mantissa του υπολογιστή σας. Να ενσωματωθούν τα υπο-προγράμματα στο πρόγραμμα της άσκησης 3 και να εκτελεσθούν οι παρακάτω πράξεις.

α) x+y+z β) x:z γ) xy:z και δ) y:(xz)

όπου x=0.234765257 10^4, y=0.213255689 10^{-2}, z=0.251434395 10^1

με ν=4.

5. Δίνεται το ολοκλήρωμα του ημιτόνου

$$\int_0^x \frac{\sin t}{t}\,dt$$

Να γραφεί ένα πρόγραμμα που να υπολογίζει το ολοκλήρωμα για διάφορα x.

Υπόδειξη: Αν αναπτύξουμε το sint σε σειρά Taylor και ολοκληρώσουμε κατά όρο βρίσκουμε

$$s(x) = x - \frac{x^3}{3.3!} + \frac{x^5}{5.5!} - \frac{x^7}{7.7!} + \frac{x^9}{9.9!} - \cdots$$

Να βρεθεί μια κατάλληλη υπολογιστική έκφραση έτσι ώστε να μην έχουμε σφάλματα υπερχείλισης. Ο τελευταίος όρος της σειράς Taylor να είναι αυτός που θα έχει απόλυτη τιμή μικρότερη από 10^{-6}. Για τον έλεγχο του προγράμματος δίνονται ορισμένα αποτελέσματα

x	S(x)
1	0.945083070
5	1.549931245
10	1.658347594

5. Θέλουμε να υπολογίσουμε τους διωνυμικούς συντελεστές που δίνονται από

$$p = \binom{m}{n}\frac{m!}{n!\,(m-n)!}$$

Να βρεθεί μια κατάλληλη υπολογιστική έκφραση έτσι ώστε να μην έχουμε σφάλμα υπερχείλισης και να γραφεί το αντίστοιχο πρόγραμμα. Για τον έλεγχο του προγράμματος δίνεται για m=35 7. Συχνά στη Στατιστική απαντάται ο διωνυμικός τύπος της πιθανότητας

$$p = \binom{m}{n}\tilde{n}^{n}(1-\tilde{n})^{m-n} \quad 0 < n < m,\ 0 < \tilde{n} < 1$$

Να βρεθεί μια υπολογιστική έκφραση και να γραφεί το αντίστοιχο πρόγραμμα έτσι ώστε να μην έχουμε σφάλματα υπερχείλισης. Να ελεγχθεί το πρόγραμμα για m=200, n=100 και ρ=0.85 ($P=3.22^{-31}$)

8. Να γραφούν δύο προγράμματα που να υπολογίζουν το άθροισμα

$$s = \frac{1}{j(j+1)} \quad j = 1,2,\dots,n$$

για διάφορες τιμές το n πρώτα με αύξουσα σειρά των όρων και μετά με φθίνουσα σειρά. Να συγκριθούν τα αποτελέσματα με την αληθή τιμή που είναι S=n/(n+1) και N=3, τότε P=6545.