Στέλιος Πολυχρόνης

ΑΜ 123603, Τμήμα ΘΕΣ-1

1η Γραπτή Εργασία

**Θέμα 1ο**

Τα σχόλια επεξήγησης του πρώτου θέματος βρίσκονται πάνω στον κώδικα.

**Θέμα 2ο**

Στις γραμμές 9-12 δηλώνω τις τέσσερεις συναρτήσεις που θα χρησιμοποιήσω παρακάτω για να υπολογίσουμε την περίμετρο των τεσσάρων γεωμετρικών σχημάτων. Οι εν λόγω συναρτήσεις ορίζονται παρακάτω στις γραμμές 62-92.

Στην γραμμή 22 (και γραμμή 57) χρησιμοποιώ την εντολή-βρόγχο «do-while» ώστε να επαναλαμβάνεται η διαδικασία έως ότου ο χρήστης επιλέξει «0. Έξοδος».

Στις γραμμές 23-28 τυπώνεται το μενού επιλογών του χρήστη για τα τέσσερα σχήματα και την επιλογή εξόδου.

Στην 29 ζητάει την επιλογή και στην 30 την διαβάζει.

Στις γραμμές 32-35, εάν ο χρήστης επιλέξει «1. Τετράγωνο», ζητάει την πλευρά του τετραγώνου, την διαβάζει, καλεί την συνάρτηση «perimtetragwnou” και τυπώνει το μήνυμα «Η Περίμετρος του τετραγώνου είναι:» μαζί με το αποτέλεσμα που επιστρέφει η συνάρτηση.

Στις γραμμές 38-41, εάν ο χρήστης επιλέξει «2. Παραλληλόγραμμο», ζητάει τις πλευρές του παραλληλογράμμου, τις διαβάζει, καλεί την συνάρτηση «perimparallilogrammou” και τυπώνει το μήνυμα «Η Περίμετρος του παραλληλογράμμου είναι:» μαζί με το αποτέλεσμα που επιστρέφει η συνάρτηση.

Στις γραμμές 44-47, εάν ο χρήστης επιλέξει «3. Ορθογώνιο Τρίγωνο», ζητάει τις κάθετες πλευρές του τριγώνου, τις διαβάζει, καλεί την συνάρτηση «perimtrigwnou” και τυπώνει το μήνυμα «Η Περίμετρος του ορθογωνίου τριγώνου είναι:» μαζί με το αποτέλεσμα που επιστρέφει η συνάρτηση.

Στις γραμμές 50-53, εάν ο χρήστης επιλέξει «4. Κύκλος», ζητάει την ακτίνα του κύκλου, την διαβάζει, καλεί την συνάρτηση «perimkuklou” και τυπώνει το μήνυμα «Η Περίμετρος του κύκλου είναι:» μαζί με το αποτέλεσμα που επιστρέφει η συνάρτηση.

Στις γραμμές 62-67 ορίζεται η συνάρτηση “perimtetragwnou”, που υπολογίζει την περίμετρο του τετραγώνου 4\*a και επιστρέφει την τιμή της.

Στις γραμμές 70-75 ορίζεται η συνάρτηση “perimparallilogrammou”, που υπολογίζει την περίμετρο του παραλληλογράμμου 2\*(n+m) και επιστρέφει την τιμή της.

Στις γραμμές 78-84 ορίζεται η συνάρτηση “perimtrigwnou”, που υπολογίζει πρώτα την υποτείνουσα c και μετά την περίμετρο του τριγώνου n+m+c και επιστρέφει την τιμή της.

Στις γραμμές 87-90 ορίζεται η συνάρτηση “perimkuklou”, που υπολογίζει την περίμετρο του κύκλου 2\*3,14159\*n και επιστρέφει την τιμή της.

**Θέμα 3ο**

Στις γραμμές 10-12 δηλώνω τις σταθερές και τις μεταβλητές του προγράμματος.

Στις γραμμές 14-15 τυπώνει την επικεφαλίδα.

Στην γραμμή 17 (και στην 23) χρησιμοποιώ την εντολή-βρόγχο «do-while» ώστε όσο ο χρήστης δίνει αρνητικό αριθμό, το πρόγραμμα ξαναζητά νέο αριθμό και τον διαβάζει. Ταυτόχρονα τυπώνει και το αντίστοιχο μήνυμα λάθους.

Όταν ο χρήστης δώσει θετικό ακέραιο αριθμό, τερματίζεται ο βρόγχος και τυπώνεται η απάντηση «Η Αντίστροφη Δυαδική Αναπαράσταση του \_\_ είναι:»

Στη γραμμή 27 ξεκινά ο βρόγχος «while» που υπολογίζει την αντίστροφη δυαδική αναπαράσταση ως εξής:

1. υπολογίζεται το υπόλοιπο της διαίρεσης ακέραιος/d (όπου d=2 σταθερά)
2. αν το υπόλοιπο είναι 0 τότε στον δυαδικό τυπώνεται ο χαρακτήρας «0», αλλιώς τυπώνεται ο χαρακτήρας «1»
3. ο ακέραιος παίρνει την νέα του τιμή ίση με το πηλίκο της διαίρεσης ακέραιος/d

και η διαδικασία αυτή επαναλαμβάνεται έως ότου ο ακέραιος γίνει ίσος με 0.

Το πρόγραμμα κλείνει όταν ολοκληρωθεί όλη η διαδικασία και τυπωθεί στην οθόνη η αντίστροφη δυαδική αναπαράσταση του ακεραίου.

**Θέμα 4ο**

Στην γραμμή 11 δηλώνω το αρχείο εισόδου fin και το αρχείο εξόδου fout.

Στην γραμμή 12 δηλώνω έναν πίνακα χαρακτήρων 7 θέσεων για την πινακίδα του λεωφορείου.

Στις γραμμές 13-14 δηλώνω τις μεταβλητές του προγράμματος και έναν πίνακα ακεραίων 53 θέσεων για τις θέσεις του λεωφορείου.

Στην γραμμή 15 θέτουμε όλα τα στοιχεία του πίνακα theseis αρχικοποιούνται ίσα με 0 ώστε όλες οι θέσεις να είναι κενές.

Στην γραμμή 17 ανοίγουμε και αντιστοιχούμε στο αρχείο εισόδου fin το αρχείο «bus.txt» μόνο για ανάγνωση.

Στην γραμμή 18 ανοίγουμε και αντιστοιχούμε στο αρχείο εξόδου fout το αρχείο «layout.txt» μόνο για εγγραφή.

Στις γραμμές 19-28 γίνεται έλεγχος εάν τα αρχεία εισόδου και εξόδου είναι κενά. Αν συμβαίνει αυτό τυπώνει μήνυμα λάθους στο άνοιγμα του αρχείου και τερματίζει το πρόγραμμα. Αλλιώς, αν όλα είναι καλά, ξεκινά το πρόγραμμα «Σύστημα Καταγραφής Θέσεων Λεωφορείου», διαβάζει από το αρχείο εισόδου και αντιστοιχεί την σειρά των 7 πρώτων χαρακτήρων στον πίνακα χαρακτήρων «pinakida» και τον αριθμό που ακολουθεί τον αντιστοιχεί στην μεταβλητή «arithm» για τον αριθμό των θέσεων.

Στις γραμμές 30-34 υπολογίζεται η μεταβλητή Ν για τις σειρές του λεωφορείου και η μεταβλητή upo που στην ουσία τσεκάρει αν ο αριθμός των σειρών είναι ακέραιο πολλαπλάσιο του 4. Οπότε η εντολή if ελέγχει αν ο αριθμός των θέσεων είναι μεγαλύτερος του 53 ή αν ο αριθμός Ν είναι αρνητικός ή αν το upo είναι διάφορο του μηδενός. Αν συμβαίνει κάτι από τα τρία, τότε ο αριθμός των θέσεων δεν συμβαδίζει με την κατανομή που θέλουμε, τυπώνεται μήνυμα λάθους και τερματίζεται το πρόγραμμα.

Στην γραμμή 36, με την εντολή do-while (γραμμή 227), ξεκινά το βασικό κομμάτι του προγράμματος, το οποίο τερματίζεται όταν ο χρήστης επιλέξει «0. Exodos». Αν και δεν το ζητάει ευθαρσώς το θέμα 4, επέλεξα το βασικό κομμάτι του προγράμματος να εκτελείται μέσα σε βρόγχο, ώστε ο χρήστης να έχει την δυνατότητα να χρησιμοποιήσει πολλές φορές και διαδοχικά τις επιλογές που του δίνει το πρόγραμμα. Έτσι μπορεί να κάνει κράτηση σε όσες θέσεις θέλει, να ακυρώσει, να δει τις κρατήσεις, να εμφανίσει την λίστα και το διάγραμμα των θέσεων, και όλα αυτά μπορούν να γίνουν όσες φορές επιθυμεί ο χρήστης. Αν όμως τερματιστεί το πρόγραμμα και ξαναξεκινήσει οι θέσεις αδειάζουν και είναι όλες κενές.

Στις γραμμές 37-49 τυπώνεται στην οθόνη η λίστα επιλογών, ζητείται από τον χρήστη να δώσει τον αριθμό της επιλογής του, διαβάζει την επιλογή και αναθέτει την δοθείσα τιμή στην μεταβλητή epilogi.

Στις γραμμές 51-62 αν η επιλογή είναι 1, τότε «διαβάζει» τις θέσεις του πίνακα theseis και «μετράει» πόσες είναι κενές και τυπώνει το σύνολό τους μαζί με το κατάλληλο μήνυμα. Ακόμα τυπώνει τους αριθμούς των κενών θέσεων.

Στις γραμμές 66-84 αν η επιλογή είναι 2, τότε ζητάει από τον χρήστη τον αριθμό της θέσης που θέλει να κρατήσει. Διαβάζει αυτόν τον αριθμό. Ελέγχει πρώτα αν είναι εντός ορίων και αν όχι τυπώνει κατάλληλο μήνυμα και ξεκινά τον βρόγχο do-while από την αρχή (k=1). Έπειτα ελέγχει αν η θέση είναι κενή και αν όχι τυπώνει κατάλληλο μήνυμα και ξεκινά τον βρόγχο do-while από την αρχή (k=1). Τέλος κρατάει τη θέση που επέλεξε ο χρήστης θέτοντας το αντίστοιχο στοιχείο στον πίνακα με την τιμή 1 και τυπώνει μήνυμα ότι η κράτηση έγινε. Η μεταβλητή k υπάρχει για να ελέγχει αν ο βρόγχος do-while πρέπει να συνεχιστεί ή όχι.

Στις γραμμές 86-118 αν η επιλογή είναι 3, τότε ξεκινάει με την εντολή for μια επανάληψη που ελέγχει όλες τις θέσεις (από 0 έως 52) του πίνακα theseis[53] και η διαδικασία έχει ως εξής:

Αν ο αριθμός i της θέσης του πίνακα (π.χ. i=0, 4, 8, … κ.τ.λ.) διαιρείται ακριβώς με το 4 και δίνει υπόλοιπο 0, και η θέση αυτή είναι κενή, τότε γίνεται κράτηση της αντίστοιχης θέσης του λεωφορείου, δηλαδή αυτής που έχει αριθμό i+1, και τυπώνεται κατάλληλο μήνυμα.

Ή αν ο αριθμός i της θέσης του πίνακα μειωμένος κατά 3, δηλαδή ο i-3 (π.χ. i=3, 7, 11, … κ.τ.λ.) διαιρείται ακριβώς με το 4 και δίνει υπόλοιπο 0, και η θέση αυτή είναι κενή, τότε γίνεται κράτηση της αντίστοιχης θέσης του λεωφορείου, δηλαδή αυτής που έχει αριθμό i+1, και τυπώνεται κατάλληλο μήνυμα. Εξαιρείται η τιμή i=51 γιατί η αντίστοιχη θέση του λεωφορείου με αριθμό 52 βρίσκεται στη γαλαρία και δεν έχει παράθυρο.

Ή αν η θέση του πίνακα i=48, δηλαδή η θέση του λεωφορείου 49, είναι κενή γίνεται κράτησή της και τυπώνεται κατάλληλο μήνυμα.

Ή αν η θέση του πίνακα i=52, δηλαδή η θέση του λεωφορείου 53, είναι κενή γίνεται κράτησή της και τυπώνεται κατάλληλο μήνυμα.

Αλλιώς αν και η τελευταία θέση i=52, δηλαδή 53, είναι κατειλημμένη τυπώνεται κατάλληλο μήνυμα ότι «δεν υπάρχει κενή θέση σε παράθυρο».

Η διαδικασία αυτή βρίσκει και κάνει κράτηση της πρώτης κενής θέσης σε παράθυρο.

Στις γραμμές 121-139 αν η επιλογή είναι 4, τότε ξεκινάει ένας βρόγχος do-while που η επανάληψή του ελέγχεται από τη μεταβλητή k. Στην αρχή εκτυπώνεται μήνυμα που ζητάει τον αριθμό της θέσης που θέλει ο χρήστης να ακυρώσει. Διαβάζει την τιμή που δίνει ο χρήστης και την αναθέτει στην μεταβλητή kratisi. Στην αρχή ελέγχει αν η τιμή είναι εντός ορίων και αν όχι τυπώνει κατάλληλο μήνυμα και ξεκινά τον βρόγχο do-while από την αρχή (k=1). Έπειτα ελέγχει αν η θέση είναι κενή και αν ναι τυπώνει κατάλληλο μήνυμα και ξεκινά τον βρόγχο do-while από την αρχή (k=1). Τέλος ακυρώνει τη θέση που επέλεξε ο χρήστης θέτοντας το αντίστοιχο στοιχείο στον πίνακα με την τιμή 0, τυπώνει μήνυμα ότι η κράτηση ακυρώθηκε και τερματίζει τον βρόγχο (k=0).

Στις γραμμές 141-158 αν η επιλογή είναι 5, τότε ξεκινάει ένας βρόγχος do-while που η επανάληψή του ελέγχεται από τη μεταβλητή k. Στην αρχή εκτυπώνεται μήνυμα που ζητάει τον αριθμό της θέσης που θέλει ο χρήστης να ελέγξει αν είναι κρατημένη. Διαβάζει την τιμή που δίνει ο χρήστης και την αναθέτει στην μεταβλητή kratisi. Στην αρχή ελέγχει αν η τιμή είναι εντός ορίων και αν όχι τυπώνει κατάλληλο μήνυμα και ξεκινά τον βρόγχο do-while από την αρχή (k=1). Έπειτα ελέγχει αν η θέση είναι κενή και αν ναι τυπώνει κατάλληλο μήνυμα και τερματίζει τον βρόγχο do-while (k=0). Τέλος ελέγχει αν η θέση είναι κρατημένη και αν ναι τυπώνει κατάλληλο μήνυμα και τερματίζει τον βρόγχο do-while (k=0).

Στις γραμμές 160-171 αν η επιλογή είναι 6, τότε «διαβάζει» τις θέσεις του πίνακα theseis και «μετράει» πόσες είναι κρατημένες και τυπώνει το σύνολό τους μαζί με το κατάλληλο μήνυμα. Ακόμα τυπώνει λίστα με ταξινομημένους τους αριθμούς των κρατημένων θέσεων.

Στις γραμμές 175-194 αν η επιλογή είναι 7, τότε τυπώνει την πινακίδα κυκλοφορίας και το διάγραμμα του λεωφορείου. Η διαδικασία για την εκτύπωση του διαγράμματος γίνεται με δύο «φωλιασμένες» εντολές for (γραμμή 179 και γραμμή 184) μέσα σε άλλη εντολή for (γραμμή 178). Η «εξωτερική» με την μεταβλητή i «σαρώνει» τις σειρές των θέσεων του λεωφορείου, και οι δύο «φωλιασμένες» με την μεταβλητή j σαρώνει τις στήλες των θέσεων του λεωφορείου. Στη διάρκεια αυτής της «σάρωσης» όπου εντοπίζει κενή θέση τυπώνει «\_», όπου εντοπίζει κρατημένη θέση τυπώνει «\*» και για τον διάδρομο ανάμεσα στην 2η και 3η στήλη τυπώνει κενό « ». Στις πέντε τελευταίες θέσεις της γαλαρίας κάνει ξεχωριστή σάρωση χωρίς να αφήνει το κενό του διαδρόμου.

Στις γραμμές 196-216 αν η επιλογή είναι 8, τότε κάνει την ίδια διαδικασία με την επιλογή 7 απλά αντί για την οθόνη, τυπώνει την πινακίδα και το διάγραμμα του λεωφορείου στο αρχείο εξόδου fout (layout.txt). Παράλληλα εμφανίζει στην οθόνη μήνυμα ότι «Το διάγραμμα αποθηκεύτηκε σε αρχείο».

Στις γραμμές 218-221 αν η επιλογή είναι 0, τότε κλείνει το αρχείο εισόδου fin και το αρχείο εξόδου fout, τυπώνει κατάλληλο μήνυμα για την «Έξοδο του Προγράμματος» και το τερματίζει σταματώντας τον αρχικό βρόγχο do-while (γραμμές 36 και 227).

Τέλος στις γραμμές 223-224 αν η επιλογή δεν είναι καμία από τις παραπάνω, τότε τυπώνει στην οθόνη μήνυμα ότι η επιλογή του χρήστη δεν είναι έγκυρη και ξαναδίνει την λίστα επιλογών ζητώντας νέα επιλογή.