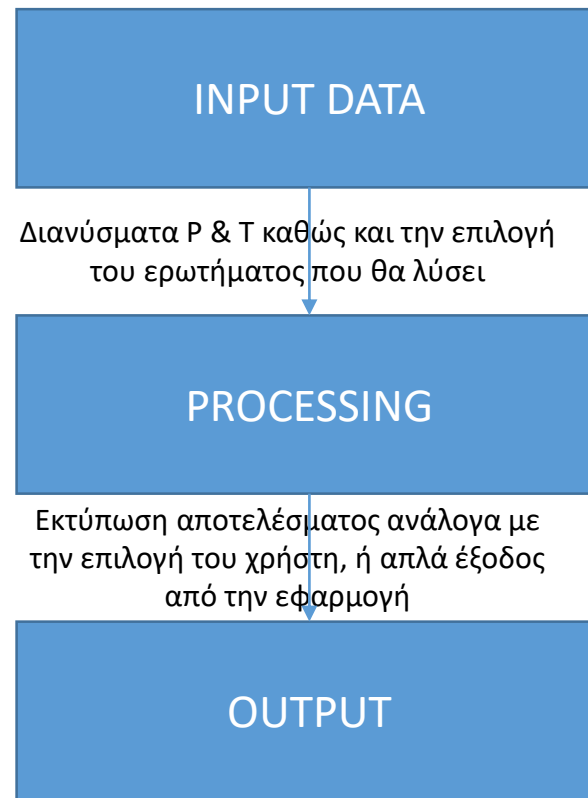


Ανάλυση Πρακτικό 5: Main



Ανάλυση Πρακτικό 5: Main

INPUT DATA functions:
loadData

P vector & T vector

Διαβάζω από το αρχείο data.txt τις τιμές πίεσης και θερμοκρασίας και τις αποθηκεύω σε δυο vectors, το P & T αντίστοιχα.

User choice:

- 1: call function choice1
- 2: call function choice2
- 3: call function choice3

Ζητάω από το χρήστη να μου επιλέξει ποιά ερώτημα της άσκησης θέλει να τρέξει, το 1 ή το 2 ή το 3. Αν πατήσει 0 τότε έξοδος από το πρόγραμμα.

Ανάλυση Πρακτικό 5: Choice1

Ο χρήστης έχει επιλέξει το ερώτημα 1: τότε τρέχω τη συνάρτηση Choice1 η οποία έχει input τα P,T vectors.

Ρωτάω το χρήστη για ποιά πίεση θέλει να υπολογίσει τη θερμοκρασία (Puser)

Ψάχνω όλες τις τιμές του P από τη θέση 1-10 για να βρω το διάστημα στο οποίο περιέχεται η τιμή P.

Έτσι έχω υπολογίσει ότι $P(k) \leq P \leq P(k+1)$, όπου k μια τιμή από το 1-9.

Αν θέσω $P1 = P(k)$, $P2 = P(k+1)$, $T1 = T(k)$ & $T2 = T(k+1)$ τότε μπορώ να βρω την τιμή του Tcalc που αντιστοιχεί για Puser χρησιμοποιώντας τη μέθοδο γραμμικής παρεμβολής.

Η μέθοδο γραμμικής παρεμβολής ορίζει ότι αν γνωρίζω τα σημεία $A1 = (P1, T1)$ & $A2 = (P2, T2)$, τότε υποθέτω ότι για μια ενδιάμεση τιμή P η θερμοκρασία θα μπορεί να υπολογιστεί από μια ευθεία ($y=a*x+b$) που ενώνει τα σημεία A1 & A2, όπου $P1 \leq P \leq P2$.

Για να υπολογίσω τα a & b της ευθείας λύνω το σύστημα εξισώσεων

$$T1 = a * P1 + b$$

$$T2 = a * P2 + b$$

Ανάλυση Πρακτικό 5: Choice1

Η λύση του προβλήματος αυτού είναι:

$a = (T2 - T1) / (P2 - P1)$, για $P1$ διάφορο του $P2$

$b = T2 - a * P2$.

Άρα το function της γραμμικής παρεμβολής θα χρειάζεται σαν παράμετρους $P1$, $P2$, $T1$, $T2$ & $Puser$

Έτσι στην μέθοδο choice1, θα χρειαστώ δυο functions. Η πρώτη function θα μου βρίσκει τις δυο διαδοχικές πειραματικές τιμές $P1$, $P2$ και τις θερμοκρασίες $T1$ & $T2$ που αντιστοιχούν. Την function αυτή θα την ονομάσω **function[P1 T1 P2 T2] = FindRange(Pvalue, P, T)**, όπου P , T τα vectors των πιέσεων – θερμοκρασιών αντίστοιχα.

Η δεύτερη function θα λύνει τη γραμμική παρεμβολή, και θα την ονομάσω **function Tcalc = LinInter(P1, P2, T1, T2, Puser)**.

Τέλος, αν το $|Puser - P1| < tol$ ή $|Puser - P2| < tol$, όπου $tol = 1e-3$, τότε θεωρώ ότι ο χρήστης έχει πληκτρολογήσει πίεση που αντιστοιχεί σε πειραματικά δεδομένα. Άρα $Tcalc = T1$ ή $Tcalc = T2$ αντίστοιχα. Αλλιώς, υπολογίζω $Tcalc$ με γραμμική παρεμβολή για πίεση $Puser$ φραγμένη από $P1$ & $P2$.

Εκτύπωση του αποτελέσματος και γνωστοποίηση ότι η τιμή έχει υπολογιστεί ή είναι πειραματική.

Ανάλυση Πρακτικό 5: Choice2

Ο χρήστης έχει επιλέξει το ερώτημα 2: τότε τρέχω τη συνάρτηση Choice2 η οποία έχει input τα P,T vectors.

Για κάθε μια τιμή του vector P εκτυπώνω την τιμή της θερμοκρασίας T.

Ανάλυση Πρακτικό 5: Choice3

Ο χρήστης έχει επιλέξει το ερώτημα 3: τότε τρέχω τη συνάρτηση Choice3 η οποία έχει input τα P,T vectors.

Εδώ ζητάει να φτιάξουμε έναν πίνακα που θα έχει τιμές πιέσεων – θερμοκρασιών υπολογισμένες με γραμμική παρεμβολή, που θα ξεκινούν από μια τιμή rangeFrom έως rangeTo με βήμα step.

Τι κάνω:

Υπολογίζω το vector με τις τιμές της πίεσης που θέλω να υπολογίσω Pvalues από rangeFrom έως rangeTo. Για να υπολογίσω όμως τις τιμές των θερμοκρασιών που αντιστοιχούν, θα πρέπει να τις υπολογίσω με γραμμική παρεμβολή.

Έτσι για κάθε τιμή Pvalue θα πρέπει να βρω τις τιμές Pexpr1 & Pexpr2 που είναι το άνω και κάτω φράγμα της. Για να υπολογίσω το άνω/κάτω φράγμα καλώ τη συνάρτηση FindRange, που μας επιστρέφει και τα Pexpr1, Pexpr2, Texpr1 & Texpr2.

Για τις τιμές αυτές καλώ τη συνάρτηση LinInter(Pexpr1, Pexpr2, Texpr1, Texpr2, P) η οποία μας επιστρέφει την θερμοκρασία που αντιστοιχεί T

Αν επαναλάβουμε για όλες τις τιμές μέχρι το rangeTo θα έχουμε τον πίνακα που μας ζητάει το ερώτημα 3.

Ανάλυση Πρακτικό 5: Κώδικας – main.m

```
clc;
clear;

disp("Welcome to calculator");

[P T] = loadData("data.txt");

disp("I have read data.txt");

choice = -1;
while true
    choice = input("Give a choice: press 1 for calculate specific T value for P, press 2 to see the table, press 3 for customized table, press 0 to exit \r\n");
    if (choice == 1)
        choice1(P, T)
    elseif (choice == 2)
        choice2(P, T)
    elseif (choice == 3)
        choice3(P, T)
    elseif (choice == 0)
        break;
    end
end

disp("Program calculator terminated");
```

Ανάλυση Πρακτικό 5: Κώδικας – loadData.m

```
function [P T] = loadData(fileName)
    clear A;
    A = importdata(fileName);
    P = A(:,1);
    T = A(:,2);
    clear A; %Free allocated memory for matrix A
end
```

Ανάλυση Πρακτικό 5: Κώδικας – choice1.m

```
function choice1(P, T)
    report = '';
    tol = 1e-3;
    Puser = input("Give me pressure to calculate temperature");
    [P1 T1 P2 T2] = FindRange(Puser, P, T);

    if ((abs(Puser - P1) < tol) | (abs(Puser - P2) < tol))
        if (abs(Puser - P1) < tol)
            Tcalc = T1;
        else
            Tcalc = T2;
        end
        report = "experiment";
    else
        Tcalc = linInter(P1, P2, T1, T2, Puser);
        report = "calculated";
    end

    fprintf("For given P = %f the temperature is T = %f and T value is %s", Puser, Tcalc, report);
end
```

Ανάλυση Πρακτικό 5: Κώδικας – FindRange.m

```
function[P1 T1 P2 T2] = FindRange(Pvalue, P, T)
    P1 = 0;
    P2 = 0;
    T1 = 0;
    T2 = 0;
    for i = 1:length(P) - 1
        if ((P(i) <= Pvalue) & (Pvalue <= P(i+1)))
            P1 = P(i);
            P2 = P(i+1);
            T1 = T(i);
            T2 = T(i+1);
            break;
        end
    end
end
```

Ανάλυση Πρακτικό 5: Κώδικας – linInter.m

```
function [T] = linInter(P1, P2, T1, T2, P)
    a = (T2 - T1) / (P2 - P1);
    b = T1 - a * P1;
    T = a * P + b;
end
```

Ανάλυση Πρακτικό 5: Κώδικας – choice2.m

```
function choice2(P, T)
    fprintf("Pressure Temperature \r\n");
    fprintf("----- \r\n");
    for i = 1:length(P)
        fprintf("%f - %f \r\n", P(i), T(i));
    end
end
```

Ανάλυση Πρακτικό 5: Κώδικας – choice3.m

```
function choice3(P, T)
    step = input("Give table step \r\n");
    rangeFrom = input("Give from P value \r\n");
    rangeTo = input("Give to P value \r\n");

    Pvalues = [rangeFrom:step:rangeTo];
    Tvalues = zeros(length(Pvalues), 1); %preallocation of Tvalues

    for i = 1: length(Pvalues) - 1
        [P1exp T1exp P2exp T2exp] = FindRange(Pvalues(i), P, T);
        Tvalues(i) = linInter(P1exp, P2exp, T1exp, T2exp, Pvalues(i));
    end

    % calculate for the last value of Pvalues
    [P1exp T1exp P2exp T2exp] = FindRange(Pvalues(end), P, T);
    Tvalues(end) = linInter(P1exp, P2exp, T1exp, T2exp, Pvalues(end));

    % to print, just call choice2 with Pvalues and Tvalues argument:
    choice2(Pvalues, Tvalues);
end
```