

Π. Σταθοπούλου

psth@ece.upatras.gr ή
psth@upatras.gr

Ομάδα Α' (Φοιτητές με μονό αριθμό Μητρώου)

- Διδασκαλία : Παρασκευή 11πμ-13μμ ΗΛ7
- Φροντιστήριο : Δευτέρα 11πμ-12πμ ΗΛ4

3ο εργαστήριο

Τι θα υλοποιήσουμε ?

Θα αναπτύξουμε ένα πρόγραμμα που θα εκτελεί τις **4 βασικές μαθηματικές πράξεις** σε ένα σύνολο αριθμών.

δηλαδή  **ένα calculator**

πως ?

Εισάγοντας αριθμητικές εκφράσεις
αντίστροφης Πολωνικής σημειογραφίας

3ο εργαστήριο

Εισαγωγή σε βασικές έννοιες

Ο τρόπος χειρισμού των δεδομένων στην γλώσσα C βασίζεται στην αντίληψη **των αφαιρετικών διαδικασιών (data abstraction)**

Με **αφαιρετικές διαδικασίες** κατασκευάζουμε **δομές** οι οποίες μας επιτρέπουν να αντιληφθούμε τα **δεδομένα** μας.

3ο εργαστήριο

Εισαγωγή σε βασικές έννοιες

Οι **δομές δεδομένων** (data structures) χρησιμοποιούνται στον προγραμματισμό επειδή μας επιτρέπουν



προσπέλαση & επεξεργασία



δεδομένων με σχετικά εύκολο τρόπο

3ο εργαστήριο

Εισαγωγή σε βασικές έννοιες

Υπάρχουν διάφοροι τρόποι να δομήσουμε τα δεδομένα μας.

Δομή Δεδομένων	Μορφή
Πίνακας (array)	ενσωματωμένη
Εγγραφή (record)	ενσωματωμένη
Συνδεδεμένος κατάλογος (linked list)	δημιουργείται
Στοιβά (stack)	δημιουργείται
Ουρά (queue)	δημιουργείται
Γραφική παράσταση (graph)	δημιουργείται
Δένδρο (tree)	δημιουργείται

3ο εργαστήριο

Εισαγωγή σε βασικές έννοιες

Στοιβά (Stack) ονομάζουμε μια δομή στην οποία τα δεδομένα προσπελάζονται με την λογική της **LIFO** (*Last in, First out*)

Η Στοιβά δεν πρέπει να συγχέεται με την ομώνυμη περιοχή της μνήμης, παρότι και οι δύο λειτουργούν με παρόμοιο τρόπο.

3ο εργαστήριο

Εισαγωγή σε βασικές έννοιες

Οι βασικές διαδικασίες που χρησιμοποιούνται για τον χειρισμό των δεδομένων μιας **Στοίβας** είναι:

ώθηση (push) & εξαγωγή (pop)

push() → εισάγει ένα δεδομένο στην κορυφή της στοίβας

pop() → εξάγει το πρόσφατο δεδομένο από την κορυφή

3ο εργαστήριο

Τι υλοποιούμε ?

Στόχος του εργαστηρίου

σχεδιασμός & ανάπτυξη ενός **calculator**

πως ?

Χρησιμοποιώντας

- * αντίστροφη Πολωνική σημειογραφία
- * διαχείριση των δεδομένων με την δομή της στοίβας

3ο εργαστήριο

Τι είναι όμως

αντίστροφη Πολωνική σημειογραφία ?

$3 + 4$
Prefix notation
Infix notation
Postfix notation

ενθεματική

$+ 3 4$
Prefix notation
Infix notation
Postfix notation

προθεματική

$3 4 +$
Prefix notation
Infix notation
Postfix notation

μεταθεματική

3ο εργαστήριο

Γενικές Οδηγίες

* Διαδικασία εκτέλεσης

Ακολουθούμε τις οδηγίες του βιβλίου (κεφ. 8 σελίδες 258 ~268 & 11.5 σελίδες 363 ~372) και του έντυπου από το eclass

* Παραδοτέα 2

Στο εργαστήριο

Τελικό παραδοτέο

Παραδοτέα Στο εργαστήριο δηλαδή σήμερα

1. Πρόγραμμα να διαχειρίζεται ακεραίους τελεστές
2. Πρόγραμμα να διαχειρίζεται δεκαεξαδικούς τελεστές

3ο εργαστήριο

Συστατικά του προγράμματος ver. 1

Ορισμός τυπικών βιβλιοθηκών

stdio.h, stdlib.h, ctype.h

Ορισμός συναρτήσεων

push(), pop(), getop(),
getch(), ungetch

Ορισμός μεταβλητών

double stack_buffer[]
char buffer[], char input[]

Κυρίως πρόγραμμα

Πηγαίος κώδικας για την
διαχείριση τελεστών & τελεστέων

Προετοιμασία για το 3ο εργαστήριο

εξάγει

```
double pop(void)
{
    if (sp>0)
        return stack_buf[--sp];
    else
    {
        printf("ERROR:STOIVA KENH\n");
        return 0;
    }
}
```

εισάγει

```
void push(double f)
{
    if
    (sp>=STACK_SIZE)
        return;
    else
    {
        stack_buf[sp++]=f;
        return;
    }
}
```

Προετοιμασία για το 3ο εργαστήριο

η **int getop(char input[])** είναι μία συνάρτηση με

- όρισμα ένα πίνακα τύπου χαρακτήρων **input[]** και
- επιστρεφόμενη τιμή τύπου ακεραίου

η **getop()** συνεργάζεται (καλεί) με τις 2 συναρτήσεις
int getch(void) & **void ungetch(int)**

η **int getch(void)** είναι μία συνάρτηση

που χρησιμοποιεί την **getchar()** για να επιστρέφει την τιμή ASCII κάθε αλφαριθμητικού χαρακτήρα που εισέρχεται από το πληκτρολόγιο

& τις καταχωρεί στον πίνακα χαρακτήρων **input[]**

Προετοιμασία για το 3ο εργαστήριο

η **void ungetch(int)** είναι μία συνάρτηση

που καλείται μόνο στην περίπτωση που ο αλφαριθμητικός χαρακτήρας που εισέρχεται από το πληκτρολόγιο δεν είναι ο **EOL**

Βασική λειτουργία της **ungetch()** είναι να ελέγχει
αν ο **temporary buffer** έχει γεμίσει ή όχι

τυπώνει: **The buffer is full**

Αναθέτει την τιμή **c** στο στοιχείο του **buffer** & συνεχίζει στο επόμενο στοιχείο.

```
void ungetch(int c) {  
    if(bufp>=MAXOP)  
        printf("THE BUFFER FOR THE INPUT ARRAY IS FULL\n");  
    else  
        buffer[bufp++]=c;  
    return;  
}
```

Προετοιμασία για το 3ο εργαστήριο

η `gettop()` εκτελεί 2 βασικούς ελέγχους

1. Ελέγχει αν `getch()` εισάγει χαρακτήρες κενού & του οριζόντιου `tab` και τους αγνοεί με την παρακάτω πρόταση

```
while((input[0]=c=getch())==' '||c=='\t');  
input[1]='\0';
```

Πως ?

Συσσωρεύει όλα τα κενά ή τα οριζόντια `tab` στο πρώτο στοιχείο της `input` και σταματάει μόνο όταν συναντήσει χαρακτήρες διάφορους του κενού ή `tab`. Μόλις σταματήσει τότε στέλνει και την πρόταση `input[1]='\0';` για να τελειώσει τον συγκεκριμένο έλεγχο.

Προετοιμασία για το 3ο εργαστήριο

η `gettop()` εκτελεί 2 βασικούς ελέγχους

2. Ελέγχει αν `getch()` της επιστρέφει χαρακτήρες διάφορους (`!=`) των αριθμητικών ψηφίων & της τελείας με την παρακάτω πρόταση

```
if (!isdigit(c)&& c!='.')
```

```
return c;
```

2. Στην συνέχεια εκτελεί 3 επί μέρους ελέγχους αν η μεταβλητή `c` είναι:

- αριθμητικό ψηφίο
- τελεία
- χαρακτήρας διάφορος του `EOL`

Εάν διάφορο των παραπάνω επιστρέφει τον χαρακτήρα

Προετοιμασία για το 3ο εργαστήριο

- Ελέγχει αν **getch()** εισήγαγε αριθμητικό ψηφίο με την πρόταση

```
if(isdigit(c))  
while(isdigit(input[++i]=c=getch()));
```
- Ελέγχει αν **getch()** εισήγαγε τον χαρακτήρα της τελείας με την πρόταση

```
if(c=='.')  
while(isdigit(input[++i]=c=getch()));
```
- Ελέγχει αν **getch()** εισήγαγε χαρακτήρα διαφορετικό του EOF και οποίος δε είναι ούτε τα παραπάνω (αριθμητικό ψηφίο, τελεία)

```
if(c!=EOF)  
ungetch(c);
```

Προετοιμασία για το 3ο εργαστήριο

1. Ελέγχει αν **getch()** της επιστρέφει χαρακτήρες του κενού ή του οριζόντιου tab

```
while((input[0]=c=getch())==' '||c=='\t');  
input[1]='\0';
```

```
for(;;) {  
    c=getch();  
    input[0]=c;  
    if(!(input[0]==' '||c=='\t'))  
        break;  
}  
input[1]='\0';
```

Πρόκειται για μια συνεχή επαναληπτική διαδικασία που σταματάει μόνο όταν ισχύσει ο έλεγχος του if

Προετοιμασία για το 3ο εργαστήριο

2. Ελέγχει αν **getch()** της επιστρέφει χαρακτήρες διάφορους (!=) των αριθμητικών ψηφίων & της τελείας.

```
if (!isdigit(c)&& c!='.')
```

```
return c;
```

Εάν ο χαρακτήρας που πληκτρολογήσαμε δεν είναι ψηφίο και ούτε τελεία τότε **input[++i]=c;**

Προετοιμασία για το 3ο εργαστήριο

1. Ελέγχει αν η **getch()** της επιστρέφει αριθμητικά ψηφία

```
if(isdigit(c))  
while(isdigit(input[++i]=c=getch()));
```

Εάν ο χαρακτήρας που πληκτρολογήσαμε είναι ψηφίο τότε `input[i]=c` & πάμε στο επόμενο στοιχείο της `input[i+1]`. Εάν δεν είναι ψηφίο σταματάμε την επανάληψη.

```
for (;;) {  
    c=getch();  
    input[++i]=c;  
    if (!isdigit(input[i]))  
        break;  
}
```

Προετοιμασία για το 3ο εργαστήριο

2. Ελέγχει αν η **getch()** της επιστρέφει τον χαρακτήρα της τελείας

```
if(c=='.')  
    while(isdigit(input[++i]=c=getch()));
```

Εάν ο χαρακτήρας που πληκτρολογήσαμε είναι τελεία τότε `input[i]='.'` & πάμε στο επόμενο στοιχείο της `input[i+1]`

```
for(;;) {  
    c=getch();  
    input[++i]=c;  
    if (!(input[i]=='.'))  
        break; }  
}
```

Προετοιμασία για το 3ο εργαστήριο

3. Ελέγχει αν η **getch()** επιστρέφει χαρακτήρα διάφορο του EOF

```
if(c!=EOF)  
    ungetch(c);
```

```
for(;;) {  
    ungetch(c);  
    if ( (c==EOF))  
        break; }  
}
```

Εάν ο χαρακτήρας που πληκτρολογήσαμε δεν είναι ούτε EOF τότε κάλεσε την `ungetch(c)`;

Προετοιμασία για το 3ο εργαστήριο

```
int IsHex(char ch)
{
  return (((ch >= '0') && (ch <= '9')) || ((ch >= 'A') && (ch <= 'F'))
         || ((C >= 'a') && (C <= 'f')));
}
```

Συνάρτηση ελέγχου
δεκαεξαδικών χαρακτήρων

or

```
{
  if((ch >= '0') && (ch <= '9')) ||
  ((ch >= 'A') && (ch <= 'F')) ||
  ((C >= 'a') && (C <= 'f')));
  return 1;
}
```

and

Προετοιμασία για το 3ο εργαστήριο

```
int getop(char input[])
{
  int i, c;
  while((input[0]=c=getch())!=' '||c=='\t');
  input[1]='\0';

  if (!isdigit(c)&&'.')
    return c;
  i=0;
  if(isdigit(c))
    while(isdigit(input[++i]=c=getch()));
  if(c=='.')
    while(isdigit(c=getch())); input[i]='\0';
  if(c!=EOF)
    ungetch(c); return NUMBER;}
}
```

περιττό

isdigit()

περιττό

Προετοιμασία για το 3ο εργαστήριο

```
push(atof(input));
```

```
push(strtol(input,NULL,16))
```

```
long strtol(char *,char **,int)
```

<stdio.h>

Όπου char* δείκτης της συμβολοσειράς

char** 1^ο στοιχείο μετά την μετατροπή

int βάση μετατροπής