# 11ο Πανελλήνιο Συνέδριο Καθηγητών Πληροφορικής
# Προετοιμασία μαθητών για τον Πανελλήνιο Διαγωνισμό Πληροφορικής

**Βασιλόπουλος Γεώργιος[1], Πάσχου Αικατερίνη[2]**
[1] gvasilopo@sch.gr, [2] gvkp@hotmail.com
[1] Εκπαιδευτικός Πληροφορικής ΠΕ19, Τεχνικός υπεύθυνος ΚΕΠΛΗΝΕΤ Καρδίτσας
**[2] Εκπαιδευτικός Πληροφορικής ΠΕ19, 2ο ΕΠΑΛ Καρδίτσας**

## Περίληψη

Ο Πανελλήνιος Διαγωνισμός Πληροφορικής (ΠΔΠ) διοργανώνεται στη χώρα μας από την Ελληνική Εταιρεία Επιστημόνων και Επαγγελματιών Πληροφορικής και Επικοινωνιών (ΕΠΥ) που φέτος κλίνει 29 έτη. Ωστόσο δεν είναι ευρέως γνωστός στον εκπαιδευτικό κλάδο και η συμμετοχή μαθητών σε αυτόν είναι ιδιαίτερα περιορισμένη. Στόχος του άρθρου είναι η διάθεση ενός εκπαιδευτικού υλικού για την ενίσχυση των μαθητών που θέλουν να συμμετέχουν στο διαγωνισμό. Προτείνεται η εξάσκηση με την Αριάδνη, ένα online σύστημα εξάσκησης με τη δυνατότητα υποβολής και αυτόματης αξιολόγησης λύσεων. Επίσης, ενδεικτικά θα παρουσιαστούν προβλήματα προηγουμένων ετών του διαγωνισμού και οι λύσεις τους με γραμμική πολυπλοκότητα και με τα κατάλληλα σχόλια.

**Λέξεις κλειδιά:** Πανελλήνιος Διαγωνισμός Πληροφορικής, ΠΔΠ, Προγραμματισμός

## Εισαγωγή

Είναι γενικά αποδεκτό ότι ο προγραμματισμός αναπτύσσει τη δημιουργικότητα, τη φαντασία στο σχεδιασμό, ικανότητες μεθοδολογικού χαρακτήρα, δεξιότητες αλγοριθμικής προσέγγισης, κατανόησης των προβλημάτων. Ως εκ τούτου ο διαγωνισμός πληροφορικής μπορεί να δώσει την ευκαιρία στους μαθητές να ασχοληθούν με τον προγραμματισμό καλλιεργώντας κατάλληλα τη σκέψη τους. Όμως, δυστυχώς και παρά τη σχεδόν τριακονταετή διενέργεια του διαγωνισμού, η συμμετοχή των μαθητών είναι πολύ μικρή, σε αντίθεση με άλλους μαθητικούς διαγωνισμούς, γεγονός που οφείλεται σε δύο βασικούς λόγους. Πρώτον, η ενασχόληση των μαθητών με τον προγραμματισμό στα πλαίσια του σχολικού τους προγράμματος είναι πολύ μικρή και αραιή. Δεύτερον, δεν υπάρχει ένα οργανωμένο διαδικτυακό σύστημα προετοιμασίας των μαθητών που επιθυμούν να ασχοληθούν επιπλέον με τον προγραμματισμό πέρα από τα σχολικά όρια και πλαίσια. Ένα σύστημα που θα ξεκινάει από τις βασικές γνώσεις και απλά προβλήματα και θα επεκτείνεται σε προχωρημένες γνώσεις και σύνθετα προβλήματα.

Στόχος της παρούσας εργασίας είναι η διάχυση ενός εκπαιδευτικού υλικού το οποίο μπορεί να αποτελέσει ένα σημείο αφετηρίας τόσο για το μαθητή του Λυκείου όσο και για το μαθητή Γυμνασίου που επιθυμεί να προετοιμαστεί για τον Πανελλήνιο Διαγωνισμό Πληροφορικής (ΠΔΠ). Αρχικά δίνεται έμφαση στην επεξήγηση των λύσεων (εισαγωγή σχολίων), σε προβλήματα προηγουμένων ετών του ΠΔΠ, κάτι που σήμερα απουσιάζει στο διαδίκτυο. Άλλωστε, για τους μαθητές που ξεκινούν τον προγραμματισμό είναι σημαντικό να έχουν στη διάθεση τους προγράμματα λύσεις με τις κατάλληλες επεξηγήσεις.

Στην παρούσα εργασία, γίνεται αναφορά σε εκπαιδευτικό υλικό που είναι διαθέσιμο στο Μάθημα προετοιμασίας για τον Πανελλήνιο Διαγωνισμό Πληροφορικής (D83119) που έχουμε δημιουργήσει στην πλατφόρμα eclass του ΠΣΔ και αφορά :

- παραδείγματα απλών προγραμμάτων σε C++
- προβλήματα προηγουμένων ετών της Α φάσης του διαγωνισμού
- προβλήματα προηγουμένων ετών της Β φάσης του διαγωνισμού
- προβλήματα προηγουμένων ετών για το επίπεδο του πρώτου προβλήματος της Γ φάσης του διαγωνισμού.

Το συγκεκριμένο εκπαιδευτικό υλικό είναι ιδιαίτερα χρήσιμο για :

- μαθητές που δεν έχουν τη δυνατότητα συμμετοχής στα μαθήματα προετοιμασίας που διοργανώνονται από την Ελληνική Εταιρεία Επιστημόνων και Επαγγελματιών Πληροφορικής και Επικοινωνιών (ΕΠΥ) σε Αθήνα και Θεσσαλονίκη.
- εκπαιδευτικούς που αναζητούν υλικό για τις ανάγκες του διαγωνισμού

Έπειτα, γίνεται αναφορά στο πολύτιμο σύστημα Αριάδνη που είναι διαθέσιμο στην πλατφόρμα του Διαγωνισμού Πληροφορικής της Κύπρου, το οποίο είναι ένα οργανωμένο on-line σύστημα εξάσκησης στον προγραμματισμό.

## Πανελλήνιος Διαγωνισμός Πληροφορικής (ΠΔΠ)

Φάσεις του διαγωνισμού

- Στην Α φάση, τίθεται ένα πρόβλημα κοινό για μαθητές Γυμνασίου και Λυκείου. Η υποβολή της λύσης γίνεται μέσω διαδικτύου. Η Α φάση διαρκεί περίπου 4 μήνες.
- Στην Β φάση, τίθεται ένα πρόβλημα για μαθητές Γυμνασίου και ένα άλλο πρόβλημα για μαθητές Λυκείου. Η υποβολή της λύσης γίνεται μέσω διαδικτύου. Η Β φάση διαρκεί περίπου 20 ημέρες. Στη Β φάση αξιολογούνται με υψηλή βαθμολογία οι λύσεις με χαμηλή πολυπλοκότητα (Ζήνδρος, 2017).
- Στην Γ φάση, τίθενται τρία προβλήματα κοινά για μαθητές Γυμνασίου και Λυκείου. Η Γ φάση διεξάγεται σε εργαστήριο πληροφορικής Πανεπιστημιακού ιδρύματος σε Αθήνα και Θεσσαλονίκη, σε υπολογιστές με λειτουργικό σύστημα Slax Linux χωρίς σύνδεση στο διαδίκτυο. Η Γ φάση διαρκεί 5 ώρες.

- Ακολουθεί Camp προετοιμασίας και η τελική φάση επιλογής των εθνικών ομάδων (4 μαθητές γυμνασίου, 4 μαθητές λυκείου). Το Camp διαρκεί 5 ημέρες.

Γλώσσες προγραμματισμού

Οι γλώσσες που δέχεται ο ΠΔΠ είναι οι : Pascal, C, C++ και Java.

Πλήθος επιτυχόντων μαθητών σε προηγούμενα έτη

**Πίνακας 1. Πλήθος επιτυχόντων Β φάσης**

| Νομός | Έτος 2015 | Έτος 2016 | Έτος 2017 |
|---|---|---|---|
| Αττικής | 68 | 55 | 75 |
| Θεσσαλονίκης | 23 | 16 | 15 |
| Λέσβου | 4 | 11 | 9 |
| Πιερίας | 3 | 0 | 1 |
| Λάρισας | 3 | 3 | 2 |
| Ηρακλείου | 2 | 1 | 1 |
| Αχαΐας | 1 | 2 | 1 |
| Καρδίτσας | 2 | 0 | 0 |
| Χανιών | 1 | 0 | 0 |
| Λακωνίας | 1 | 0 | 0 |
| Ιωαννίνων | 1 | 0 | 0 |
| Ημαθίας | 1 | 1 | 2 |
| Πέλλας | 1 | 0 | 1 |
| Έβρου | 1 | 1 | 0 |
| Χίου | 0 | 2 | 0 |
| Μεσσηνίας | 0 | 2 | 0 |
| Χαλκιδικής | 0 | 1 | 0 |
| Εύβοιας | 0 | 1 | 2 |
| Κορινθίας | 0 | 0 | 1 |
| Σύνολο | 112 | 96 | 110 |

**Πίνακας 2. Πλήθος επιτυχόντων Γ φάσης**

| Νομός | Έτος 2015 | Έτος 2016 | Έτος 2017 |
|---|---|---|---|
| Αττικής | 12 | 7 | 21 |
| Θεσσαλονίκης | 11 | 6 | 5 |
| Λάρισας | 2 | 2 | 1 |
| Λέσβου | 2 | 1 | 2 |
| Ηρακλείου | 0 | 0 | 1 |
| Αχαΐας | 0 | 0 | 1 |
| Σύνολο | 27 | 16 | 31 |

Σύμφωνα με τα αποτελέσματα των ετών 2015, 2016, 2017 (Πίνακας 1 και Πίνακας 2), παρατηρούμε ότι το μεγαλύτερο πλήθος επιτυχιών στις φάσεις Β και Γ του διαγωνισμού παρουσιάζονται στους νομούς Αττικής και Θεσσαλονίκης. Με την παρούσα εργασία, ευελπιστούμε να δώσουμε τα απαραίτητα εφόδια και σε μαθητές που ζουν εκτός των δύο μεγάλων αστικών κέντρων της χώρας, ώστε να μπορούν να διεκδικήσουν μια επιτυχία τουλάχιστον στη Β φάση του διαγωνισμού.

Σύστημα εξάσκησης

Στην Ελλάδα, για την εξάσκηση των μαθητών για τον ΠΔΠ, υπάρχει διαθέσιμη στο διαδίκτυο η πλατφόρμα HelleniCO (Καραγεώργος & Μόρμορης, 2012), η οποία περιέχει θεωρία χωρισμένη σε κεφάλαια και ενότητες, και προβλήματα με αυξημένη δυσκολία, γεγονός που μπορεί να αποθαρρύνει ένα νέο μαθητή που αναζητά τα πρώτα του βήματα στον προγραμματισμό πιθανώς χωρίς τη βοήθεια ενός εκπαιδευτικού. Για το λόγο αυτό, προτείνουμε το σύστημα εξάσκησης Αριάδνη.

## Σύστημα εξάσκησης Αριάδνη

Το σύστημα εξάσκησης Αριάδνη (Ολυμπιάδα πληροφορικής Κύπρου, 2017), είναι ανοιχτό σε κάθε χρήστη του διαδικτύου. Παρέχει στο χρήστη τη δυνατότητα εξάσκησης με απλά προβλήματα, χωρισμένα σε κατηγορίες και επίπεδα και δυνατότητα online υποβολής, ελέγχου ορθότητας και συγκέντρωσης πόντων. Αποτελεί ιδανικό περιβάλλον για αρχαρίους καθώς περιέχει αρκετά απλά προβλήματα (επίπεδο 0 και 1) αλλά και προβλήματα για προχωρημένους (επίπεδο 2 και 3)

Στον πίνακα 3 που ακολουθεί, αποτυπώνεται το περιεχόμενο του συστήματος Αριάδνη, το οποίο αποτελείται από 112 προβλήματα χωρισμένα σε 11 κατηγορίες.

**Πίνακας 3. Προβλήματα**

| Κατηγορία | Πλήθος | Επίπεδο |
|---|---|---|
| Ακολουθιακή Δομή | 10 | 0 |
| Δομή Διακλάδωσης | 9 | 0 / 1 |
| Δομή Επανάληψης | 18 | 0 / 1 / 2 |
| Πίνακες | 17 | 0 / 1 / 2 |
| Στοίβες | 6 | 0 / 1 / 2 / 3 |
| Γράφοι | 10 | 1 / 2 / 3 |
| Data Structures and STL | 21 | 0 / 1 / 2 |
| Divide and Conquer | 1 | 1 |
| Greedy | 2 | 2 / 3 |
| Dynamic Programming | 3 | 3 |
| Ad-Hoc | 15 | 0 / 1 / 2 / 3 |
| **Σύνολο** | **112** | |

## Ενδεικτικά προβλήματα συστήματος Αριάδνη

**Πρόβλημα 1: (Ακολουθιακή δομή, επίπεδο 0)**
Γράψτε ένα πρόγραμμα που θα διαβάζει από το πληκτρολόγιο πόσες ώρες, πόσα λεπτά και πόσα δευτερόλεπτα έχουν περάσει από τα μεσάνυχτα. Ακολούθως θα το μετατρέπει σε δευτερόλεπτα και θα τυπώνει το αποτέλεσμα στην οθόνη.

**Πρόβλημα 2: (Data Strucrures and STL, επίπεδο 2)**
Αν σας δοθούν N μη ταξινομημένοι, ακέραιοι αριθμοί, να βρείτε το ζεύγος με τη μικρότερη διαφορά στην απόλυτη τιμή της. Αν υπάρχουν περισσότερα ζεύγη να εμφανίζονται όλα.

## Ενδεικτικά προβλήματα διαγωνισμού και προετοιμασίας προηγουμένων ετών

Τα προγράμματα, λύσεις των προβλημάτων του διαγωνισμού αναρτώνται στο διαδίκτυο χωρίς τον κατάλληλο σχολιασμό, γεγονός που εφιστά την κατανόηση τους ιδιαίτερα δύσκολη όχι μόνο από τους μαθητές αλλά και από τους εκπαιδευτικούς πληροφορικής. Η μορφή αυτή των λύσεων πιθανόν αποθαρρύνει τόσο τον μαθητή όσο και τον εκπαιδευτικό από την ενασχόληση του με το διαγωνισμό. Παρακάτω παρουσιάζονται ενδεικτικά κάποια προγράμματα λύσεις εμπλουτισμένα με αρκετά σχόλια. Η μορφή αυτή κάνει την κατανόησή τους πιο εύκολη και ανατρέπει την αρχική και λανθασμένη αντίληψη για την ιδιαίτερη δυσκολία των προβλημάτων του διαγωνισμού. Οι λύσεις που ακολουθούν έχουν δοκιμαστεί τόσο με τον DevC++ compiler for Windows, όσο και με τον g++ compiler for Linux.

**Πρόβλημα 1: Σε ακολουθία θετικών ακεραίων $A_1, A_2,\ldots, A_N$ να βρεθεί το πλήθος των ζευγών $(A_i, A_j)$ όπου $1\le i\le j\le N$ και $A_i + A_j = X$.**
Το πρόβλημα είναι απλό να υλοποιηθεί με πολυπλοκότητα $O(n^2)$. Ωστόσο παρακάτω διατυπώνεται μια λύση με πολυπλοκότητα O(n).

```
// Camp 2015, Γυμνάσιο (junior), Πρόβλημα sumx
#include <fstream>
#define MAX 1000001  // ορισμός σταθεράς
/* Ο βοηθητικός πίνακας count χρησιμοποιείται για την αποθήκευση του πλήθους εμφάνισης του κάθε στοιχείου που βρίσκεται στο αρχείο εισόδου  */
int count[MAX];
using namespace std;

int main ()
{
ifstream fin("sumx.in");   // Δήλωση αρχείου εισόδου
ofstream fout("sumx.out");   // Δήλωση αρχείου εξόδου

// N=πλήθος στοιχείων,   X=αριθμός στόχος
  int N, X;
  fin >> N >> X;   // Εισαγωγή πλήθους στοιχείων και αριθμού στόχου από το αρχείο εισόδου
  int pairs = 0;   //  Η μεταβλητή pairs αποθηκεύει το ζητούμενο πλήθος
/* Αρχικά ο πίνακας count έχει σε κάθε θέση του, το στοιχείο 0.
Για κάθε στοιχείο εισόδου a, υπολογίζω την τιμή X- a, δηλαδή το συμπλήρωμα του a ως προς το στόχο X */
  for (int i=0; i<N; i++) {     // αρχή εντολής επαναψης for
    int a, b;
    // Εισαγωγή στοιχείων από το αρχείο εισόδου
    fin >> a;
```

```
  b=X-a;   // το συμπλήρωμα του a ως προς το στόχο X
   /* Εάν το b δεν είναι αρνητικό και δεν ξεπερνά το MAX, τότε το ζητούμενο πλήθος pairs αυξάνεται κατά την τιμή που είναι αποθηκευμένη στη θέση b του πίνακα count  */
   if (0 <= b && b <= MAX) pairs += count[b];

  /* Αυξάνεται κατά 1 το πλήθος εμφάνισης του αριθμού a και αποθηκεύεται στη θέση a του πίνακα count */
   count[a]++;
 }   // τέλος εντολής επανάληψης for
 fout << pairs;     // Εγγραφή πλήθους ζευγών στο αρχείο εξόδου
 return 0;
}  // τέλος προγράμματος
```

Παράδειγμα αρχείου εισόδου  sumx.in
9 13   // πλήθος στοιχείων N=9,   αριθμός στόχος X=13
5 12 7 10 9 1 2 3 11   //  τα 9 στοιχεία
Παράδειγμα αρχείου εξόδου  sumx.out
3
Επεξήγηση : Βρέθηκαν τα παρακάτω 3 ζεύγη των οποίων το άθροισμα είναι ίσο με 13.
12+1=13
10+3=13
2+11=13

**Πρόβλημα 2: Σε μονοδιάστατο πίνακα να βρεθεί εάν υπάρχει στοιχείο με όλα τα προηγούμενα στοιχεία μικρότερα και όλα τα επόμενα στοιχεία μεγαλύτερα.**
Το πρόβλημα είναι απλό να υλοποιηθεί με πολυπλοκότητα $O(n^2)$. Ωστόσο παρακάτω διατυπώνεται μια λύση με πολυπλοκότητα O(n).

```
// 26ος ΠΔΠ, Β Φάση, Λύκειο
#include <fstream>
using namespace std;
int N, a[1000009], amax[1000009], amin[1000009],
int cmin=1000009; cmax=-1, num=-1;

int main () {
ifstream fin("solar.in");  // Δήλωση αρχείου εισόδου
ofstream fout("solar.out");  // Δήλωση αρχείου εξόδου

fin >> N;    // Εισαγωγή από το αρχείο εισόδου το πλήθος N των στοιχείων
/* Ξεκινώντας από την αρχή του πίνακα, χρησιμοποιώ ένα βοηθητικό πίνακα amax για να αποθηκεύσω το 1 όταν βρίσκω στοιχείο μεγαλύτερο από όλα τα προηγούμενα. */
for (int i=1;i<=N;i++) {
        fin >> a[i];  // Εισαγωγή στοιχείων από το αρχείο εισόδου
        if (a[i]>cmax) amax[i]=1;
        else  amax[i]=0;

        // Αποθηκεύω στο cmax το μεγαλύτερο στοιχείο μέχρι στιγμής
        if (a[i]>cmax) cmax=a[i];
}
/* Ξεκινώντας από το τέλος του πίνακα, χρησιμοποιώ ένα βοηθητικό πίνακα amin για να αποθηκεύω το 1 όταν βρίσκω στοιχείο μικρότερο από όλα τα προηγούμενα. */
for (int i=N;i>=1;i--) {
        if (cmin>a[i]) amin[i]=1;
        else amin[i]=0;

        // Αποθηκεύω στο cmin το μικρότερο στοιχείο μέχρι στιγμής
        if (a[i]<cmin) cmin=a[i];
/* Στη θέση όπου οι 2 βοηθητικοί πίνακες θα έχουν την τιμή 1, βρίσκεται στοιχείο που ικανοποιεί τη συνθήκη του προβλήματος. Εάν υπάρχουν περισσότερα από ένα στοιχεία που ικανοποιούν τη συνθήκη, βρίσκω το μεγαλύτερο από αυτά.  */
        if (amax[i]==1 && amin[i]==1 && a[i]>num) num=a[i];
} // τέλος εντολής επανάληψης for
// Εγγραφή αποτελέσματος στο αρχείο εξόδου
 // Εάν το num έχει την αρχική του τιμή, τότε δεν υπάρχει στοιχείο που να ικανοποιεί τη συνθήκη του προβλήματος
if (num == -1) fout << "NOT FOUND" << endl;
else fout << num << endl;
return 0;
}  // τέλος προγράμματος
```

Παράδειγμα αρχείου εισόδου solar.in
10   // πλήθος στοιχείων N=10
3
2
4

1
5
7
8
9
10
8

Παράδειγμα αρχείου εξόδου solar.out
7

**Πρόβλημα 3 : Σε ακολουθία ακεραίων να βρεθούν 2 υποσύνολα μήκους Κ το καθένα με μέγιστο άθροισμα.**
Το πρόβλημα είναι απλό να υλοποιηθεί με πολυπλοκότητα $O(n^3)$. Ωστόσο παρακάτω διατυπώνεται μια λύση με πολυπλοκότητα O(n).

```
// 28ος ΠΔΠ, Β Φάση, Λύκειο
#include <vector>
#include <fstream>
using namespace std;
int main ()
{
  int N, K;
ifstream fin("scrabble1d.in");   // Δήλωση αρχείου εισόδου
ofstream fout("scrabble1d.out");   // Δήλωση αρχείου εξόδου

// Εισαγωγή από το αρχείο εισόδου το πλήθος N των στοιχείων και το μήκος K του υποσυνόλου
  fin >> N >>  K;
  vector<int> a(N), s(N);  // δυναμική δομή δεδομένων vector
  int i, j;
  for (i = 0; i < N; i++) fin >> a[i];
  s[0] = 0;
  for (i = 0; i < K; i++) s[0] += a[i];  // το άθροισμα της 1ης K-άδας μόνο

  for (i = K; i < N; i++) s[i-K+1] = s[i-K] - a[i-K] + a[i];
/* s[1] = s[0]-a[0]+a[K]  δηλαδή  s[1]=το άθροισμα των στοιχείων της 1ης K-άδας s[0] – 1ο στοιχείο a[0] + επόμενο στοιχείο a[K]
s[2] = s[1]-a[1]+a[K+1]  δηλαδή  s[2]= το άθροισμα των στοιχείων της 2ης K-άδας s[1] – 1ο στοιχείο a[1] + επόμενο στοιχείο
a[K+1]
κλπ. */

  int best = 0, left = 0;
/* best είναι η μεταβλητή που αποθηκεύει το ζητούμενο άθροισμα 2 υποσυνόλων,     left είναι η μεταβλητή που αποθηκεύει την πρώτη
K-άδα με μεγάλο άθροισμα στοιχείων */
  for (i = 0, j = K; j <= N-K; i++, j++) {  // εντολή επανάληψης for με 2 μεταβλητές i και j
    /* εάν βρέθηκε K-άδα με μεγαλύτερο άθροισμα στοιχείων από την προηγούμενη, τότε κράτα στη μεταβλητή left την K-άδα με το
μεγαλύτερο άθροισμα στοιχείων  */
    if (s[i] > left) left = s[i];
    if (left + s[j] > best) best = left + s[j];  // κράτα στη μεταβλητή best το μεγαλύτερο άθροισμα left+s[j]
  }  // τέλος εντολής επανάληψης for
  fout << best;     // Εγγραφή αποτελέσματος στο αρχείο εξόδου
  return 0;
} // τέλος προγράμματος
```

Παράδειγμα αρχείου εισόδου  scrabble1d.in
10 3  // *πλήθος στοιχείων N=10,  μήκος υποσυνόλου K=3*
2 4 15 12 10 1 1 20 4 10  // *τα 10 στοιχεία*
Παράδειγμα αρχείου εξόδου  scrabble1d.out
71
Επεξήγηση : Βρέθηκαν 2 υποσύνολα μήκους 3 (τριάδες) των οποίων το άθροισμα είναι ίσο με 34+37=71.
15+12+10=37
20+4+10=34

## *Συμπεράσματα*

Με την παρούσα εργασία, γίνεται μια προσπάθεια συγκέντρωσης εκπαιδευτικού υλικού με προγράμματα και επεξηγήσεις, το οποίο μπορεί να αποτελέσει ένα σημείο εκκίνησης για κάποιον μαθητή που επιθυμεί να συμμετέχει στον ΠΔΠ. Το εκπαιδευτικό υλικό που συγκεντρώνεται στο eclass του ΠΣΔ είναι ανοικτό σε κάθε χρήστη του διαδικτύου, είναι δυναμικό καθώς συνεχίζει να εμπλουτίζεται και να ενημερώνεται και αποτελεί σημείο αναφοράς για όποιον μαθητή ή εκπαιδευτικό επιθυμεί να προετοιμαστεί για τον ΠΔΠ.

## *Αναφορές*


Ζήνδρος, Δ., Μια Εύπεπτη Εισαγωγή στην Ανάλυση Πολυπλοκότητας Αλγορίθμων, Ανακτήθηκε στις 10 Φεβρουαρίου 2017 από http://discrete.gr/complexity/?el

Καραγεώργος, Π., Μόρμορης, Ε. (2012). Ο Πανελλήνιος Διαγωνισμός Πληροφορικής και οι Διεθνείς Διαγωνισμοί της ΙΟΙ, *Πρακτικά 4th Conference on Informatics in Education "Η πληροφορική στην εκπαίδευση"* (σ. 656-668), Πανεπιστήμιο Πειραιώς: Εκδόσεις Νέων Τεχνολογιών.

ΚΕ.ΠΛΗ.ΝΕ.Τ. Καρδίτσας, Προβλήματα προηγουμένων ετών, Ανακτήθηκε στις 10 Φεβρουαρίου 2017 από http://blogs.sch.gr/plinetks/contest

Μάθημα προετοιμασίας για τον Πανελλήνιο Διαγωνισμό Πληροφορικής (D83119), Ανακτήθηκε στις 10 Φεβρουαρίου 2017 από http://eclass.sch.gr/courses/D83119/

Ολυμπιάδα πληροφορικής Κύπρου, σύστημα Αριάδνη, Ανακτήθηκε στις 10 Φεβρουαρίου 2017 από http://www.coinformatics.org/

Πανελλήνιος Διαγωνισμός Πληροφορικής (ΠΔΠ), Ανακτήθηκε στις 10 Φεβρουαρίου 2017 από http://www.pdp.gr/

Πανελλήνιος Διαγωνισμός Πληροφορικής (ΠΔΠ), Σύστημα HelleniCO, Ανακτήθηκε στις 10 Φεβρουαρίου 2017 από http://www.hellenico.gr/

Πρόβλημα 26ου ΠΔΠ, Β φάση, Λύκειο. (2014), Ανακτήθηκε στις 10 Φεβρουαρίου 2017 από http://www.pdp.gr/files/26b/PDP_26_B_LYK.pdf

Πρόβλημα 28ου ΠΔΠ, Β φάση, Λύκειο. (2016), Ανακτήθηκε στις 10 Φεβρουαρίου 2017 από http://www.pdp.gr/files/PDP-28_B_LYK.pdf

Πρόβλημα Camp προετοιμασίας (2015), Γυμνάσιο, Ανακτήθηκε στις 10 Φεβρουαρίου 2017 από https://git.softlab.ntua.gr/public/pdp-camp/wikis/home

Υποστήριξη μαθημάτων για τον ΠΔΠ, Ανακτήθηκε στις 10 Φεβρουαρίου 2017 από http://www.epy.gr/wiki/index.php?title=Κεντρική_Σελίδα