

ECE 325 Practical Assignment 1

Μέλη ομάδας : Σώτος Βασιλείου , Μιχάλης Μιχαήλ , Πασιουρτίδης Κώστας

Πρόγραμμα 1:

Υλικό : [std::reverse\(\) in C++ - GeeksforGeeks](#)

Αρχικά ορίζω ένα string και δέχομαι από το χρήστη την λέξη για έλεγχο, στην συνέχεια μέσω της συνάρτησης palindrome ελέγχω και τυπώνω το ανάλογα μηνύματα.

```
string Palindrome(string S)
```

```
{
```

```
    string Palin = S;
```

```
    reverse(Palin.begin(), Palin.end()); //h sinartisti reverse antistrefi tin leksi apo to
    telos pros tin arxi diladi an exw tin leksi marios tha gini soiram
```

```
    if (S == Palin) { // elexw an meta tin antistrofi h arxiki mou leksi einai idia me tin nea
    leksi pou eGINE revers kai analogos tipono minima eksodou
```

```
        return "Palindrome";
```

```
    }
```

```
    else {
```

```
        return "Is not Palindrome";
```

```
    }
```

```
}
```

Πρόγραμμα 2:

Αρχικά δεχομαι τους αριθμους που δινει ο χρηστης π.χ. 021120 και στη συνεχεια η συναρτηση void convertStrtoArrThenSort(string str) κανει την υπολοιπη δουλια. Η συναρτηση αυτη ειναι υπευθυνη για να παρει ως string (δεχεται ως εισοδο string) τους αριθμους και να τους μετατρεψει σε integers αποθηκευοντας τους στο πινακα arr αλλα και να τους ταξινομηση στο τελος.

```
void convertStrtoArrThenSort(string str)
{
    // get length of string str
    int strlength = str.length();    //length of string
    int arr[strlength];

    int i;

    for (i = 0; i<strlength; i++)
        arr[i]=0;    //initialised the array (arr) with zero
    int j,k;
    i=0;

    for(k=0; k<strlength; k++){
        arr[k] = arr[k] * 10 + (str[i] - 48); //μετατρεπω τα string σε integers και //ταυτοχρονα τα αποθηκευω στο πινακα arr
        i++;    //αυξανω μετρητη για επομενη θεση του string
    }

    // απο κατω η συνεχεια του κωδικα

    //εδω ξεκινα το sorting οπου εχουμε τις κατηγοριες των αριθμων δηλαδη αν ειναι το 0 ή το 1 ή το 2 το οποιο το βλεπουμε με την χρηση ενός switch που εχουμε βαλει .

    Οι μεταβλητες lo και mid κανουν point στην αρχη του πινακα ενω ο hi κανει στο τελος.
```

```

int lo = 0;

int hi = strlen - 1;

int mid = 0;

while (mid <= hi) {
    switch (arr[mid]) {

        case 0:        // If the element is 0

            swap(arr[lo++], arr[mid++]);    //στην swap γινεται η ανταλλαγη των
//αριθμων σε αυτη την περιπτωση ειναι η περιπτωση του 0. Ο lo και mid ειναι
μετρητες για να αυξανονται σε επομενες θεσεις.

            break;

        case 1:        // If the element is 1 .

            mid++;

            break;

        case 2:        // If the element is 2

            swap(arr[mid], arr[hi--]);

            break;
    }
}

for (j = 0; j <= strlen-1; j++)        //τυπωση του τξινομημενου array (arr)

    cout << arr[j] << " ";
}

```

Πρόγραμμα 3

```
Struct {  
    Int Id , mid , final , total  
    char * name  
}
```

```
// synartisi gia ektipwsi
```

```
Void Ektiposi (mathiti , thesi_mathiti_gia ektiposi){  
    Printf ("kathe stixeio" , mathitis[thesi].analogo_stoixeio  
}
```

```
Void Main {
```

```
    //Orismos parametrwn  
    //Arxikopoiisi mathiton me '0'  
    Mathitis[gia olous] = 0;
```

```
Ektypwsi epilogn kai epilogi apo xristi
```

```
Case1://add student records
```

```
    SCANF( <<ID>> )
```

```
    //elegxo an ksanaiparxi  
    FOR(mexri to telos tou pinaka ){
```

```
        IF( tautotites_pou_iparxoun == tautotia_pou_pira ){
```

```
            Minima oti ksanaiparxi  
            Eksodos apo for  
        }
```

```
        IF(iparxi ksana to id )  
            Eksodos apo case
```

```
        //Kataxwrisi stoixeion apo xristi  
        Mathitis[.ola_ta_stoixeia =
```

```
        Continue;
```

```
Case2: // // delete strudent records
```

```
    SCANF(<<to ID tou mathiti pou thelw na diagrapso >>)
```

```
    FOR(gia kathe mathiti ){  
        IF( an to ID pira iparxi ){
```

```

        Mathitis.stoixeia = 0
        Arithmos_atomon -- ;
    }
    Continue ;

```

Case3: //update student records

```

SCANF(<<to ID tou mathiti pou thelw na kanw update>>)

FOR(mexri to telos tou pinaka ){

    IF(an_ I_tautotita_pou_mou_edwse_o_xristis_iparxi ){
        SCANF(<<nea_stoixeia_ston_mathiti>>)
    }

    Continue;

```

Case4: // view all student records

```

    FOR ( kathe_stoixeio_tou_pinaka ) {
        IF (an_iparxei_to_atomo){
            Sinartisi_Ektypwsi
        }
    }

    Continue;

```

Case5: //calculate the averge score of all students

```

    FOR(kathe_stoixeio_tou_pinaka_pou_Exei_iparkto_mathiti){
        Sum=sum+total[paron_mathiti]
    }
    Avarege =sum/ sinolo_iparkton_mathitwn

    PRINTF(<<avarage>>)
    Continue;

```

Case6: // show the strudent with the highest total score

```

    FOR(kathe_stoixeio_tou_pinaka)
        IF(paron_megaliteros_Apo_epomeno)
            Max_total = paron

    PRINTF(<<stoixeia max_total)

    Continue;

```

Case7: //show the student with the lowest total score

```
FOR(kathe_stoixeio_tou_pinaka)
    IF(paron_megaliteros_Apo_epomeno &&
       mexri_ekei_pou_exi_mathites )
        Max_total = paron

    PRINTF(<<stoixeia max_total)

Continue;
```

Case8: // end

```
PRINTF("TELOS ")
RETURN 0 ;
```