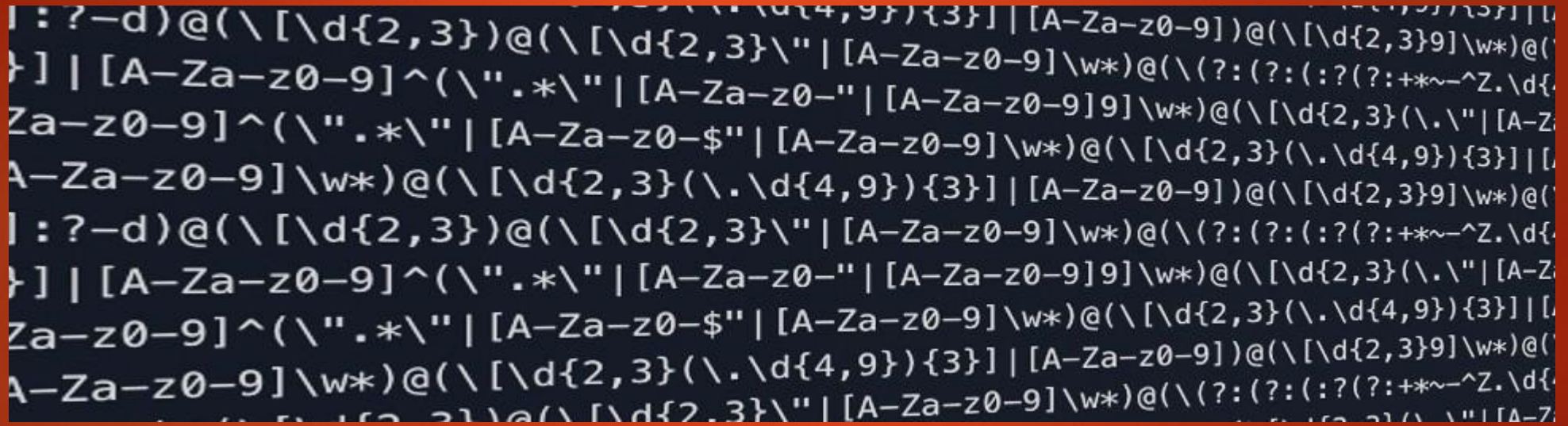
REGULAR EXPRESSIONS



Κανονικές Εκφράσεις

- Τρόπος εφαρμογής μίας ακολουθίας για εφαρμογή κανόνων γραμματικής κατά συγγραφή μίας συμβολοσειράς(string).
- Χρησιμοποιήται σε εφαρμογές όπως τα search enginees,χρήση για συγγραφή ενός έγκυρου λογαριασμού ηλεκτρονικής αλληλογραφίας,χρήση σε βάσεις δεδομένων(Π.χ εντολή like→select * from students where name like %a(Εύρεση μαθητών που το όνομα τους τελειώνει σε α)).

Βασικοί κανόνες Συγγραφής ΚΕ

- ^[]→Σύμβολο που υποδηλώνει ότι μία συμβολοσειρά ξεκινάει με κάτι.
- $→Σύμβολο που υποδηλώνει ότι μία συμβολοσειρά τελειώνει με κάτι
- *→Επανάληψη μίας ακολουθίας καμία η περισσότερες φορές.
- [^]→οποιοσδήποτε χαρακτήρας.
- [a-z]→οποιοσδήποτε πεζός λατινικός χαρακτήρας
- [A-Z]→οποιοσδήποτε κεφαλαίος λατινικός χαρακτήρας.
- A | B→Επιλογή ανάμεσα στο A η το B.
- Π.χ[^cat([^])*] →Αναγνωρίζονται λέξεις που ξεκινάνε με τα γράμματα c-a-t→Π.χ→cat,category
- Μηχανές αναγνώρισης ΚΕ→Ντετερμινιστικά πεπερασμένα αυτόματα

Χρήση κανονικών εκφράσεων στην C++

- Χρήση αντικειμένων regex.
- std::regex r(string input)→Όρισμα μία κανονική έκφραση γραμμένη σε αλφαριθμητικό.
- std::regex_match(string s,regex r).→Έλεγχος αν ένα αλφαριθμητικό ικανοποιεί κάποιους κανόνες παραγωγής.

Βασικές Κανονικές Εκφράσεις

- [a-z]*→Ορίζεται ότι τα αποδεκτά αλφαριθμητικά είναι τα αλφαριθμητικά που περιέχουν όσους πεζούς χαρακτήρες θέλω.
- [^]*→συμβολοσειρά που περιέχει οποιονδήποτε χαρακτήρα επιθυμώ όσες φορές θέλω.π.χ→AS*()IKM

ΠΑΡΑΔΕΙΓΜΑΤΑ ΣΥΜΒΟΛΟΣΕΙΡΩΝ

```
#include <iostream>
#include <regex>

int main(int argc,char **argv)
{
    if(argc!=2)
    {
        std::cerr<<"Wrong amount of arguments"<<std::endl;
        exit(EXIT_FAILURE);
    }
    std::string a="am([^])*";
    std::regex r(a);
    std::string input=argv[argc-1];
    if(std::regex_match(input,r))
    {
        std::cout<<"String:"<<input<<" matches prototype:"<<a<<std::endl;
    }
    else
    {
        std::cout<<"String:"<<input<<" does not match prototype:"<<a<<std::endl;
    }
    return 0;
}
```

Έναρξη αλφαριθμητικού με το λεκτικό amount

ΠΑΡΑΔΕΙΓΜΑΤΑ ΣΥΜΒΟΛΟΣΕΙΡΩΝ

```
#include <iostream>
#include <regex>

int main(int argc,char **argv)
{
    if(argc!=2)
    {
        std::cerr<<"Wrong amount of arguments"<<std::endl;
        exit(EXIT_FAILURE);
    }
    std::string a="am([^])*";
    std::regex r(a);
    std::string input=argv[argc-1];
    if(std::regex_match(input,r))
    {
        std::cout<<"String:"<<input<<" matches prototype:"<<a<<std::endl;
    }
    else
    {
        std::cout<<"String:"<<input<<" does not match prototype:"<<a<<std::endl;
    }
    return 0;
}
```

Έναρξη αλφαριθμητικού με το λεκτικό amount

Έλεγχος αν η συμβολοσειρά μας είναι αποδεκτή

ΠΑΡΑΔΕΙΓΜΑ ΣΥΜΒΟΛΟΣΕΙΡΩΝ(2)

```
#include <iostream>
#include <regex>

int main(int argc,char **argv)
{
    if(argc!=2)
    {
        std::cerr<<"Wrong amount of arguments"<<std::endl;
        exit(EXIT_FAILURE);
    }
    std::string a="([^])*am$";
    std::regex r(a);
    std::string input=argv[argc-1];
    if(std::regex_match(input,r))
    {
        std::cout<<"String:"<<input<<" matches prototype:"<<a<<std::endl;
    }
    else
    {
        std::cout<<"String:"<<input<<" does not match prototype:"<<a<<std::endl;
    }
    return 0;
}
```

Έλεγχος αν ένα αλφαριθμητικό τελειώνει με την λέξη am.

ΠΑΡΑΔΕΙΓΜΑ ΣΥΜΒΟΛΟΣΕΙΡΩΝ(3)

```
#include <iostream>
#include <regex>

int main(int argc,char **argv)
{
    if(argc!=2)
    {
        std::cerr<<"Wrong amount of arguments"<<std::endl;
        exit(EXIT_FAILURE);
    }
    std::string a="^a[^]a[^]a[^]";
    std::regex r(a);
    std::string input=argv[argc-1];
    if(std::regex_match(input,r))
    {
        std::cout<<"String:"<<input<<" matches prototype:"<<a<<std::endl;
    }
    else
    {
        std::cout<<"String:"<<input<<" does not match prototype:"<<a<<std::endl;
    }
    return 0;
}
```

Έλεγχος αν ένα αλφαριθμητικό ξεκινάει με α και ανα 1 ψηφίο εμφανίζεται το α.

ΠΑΡΑΔΕΙΓΜΑ ΣΥΜΒΟΛΟΣΕΙΡΩΝ(4)

```
#include <iostream>
#include <regex>

int main(int argc,char **argv)
{
    if(argc!=2)
    {
        std::cerr<<"Wrong amount of arguments"<<std::endl;
        exit(EXIT_FAILURE);
    }
    std::string a="([b-zA-Z]*a[b-zA-Z]*){5,6}";
    std::regex r(a);
    std::string input=argv[argc-1];
    if(std::regex_match(input,r))
    {
        std::cout<<"String:"<<input<<" matches prototype:"<<a<<std::endl;
    }
    else
    {
        std::cout<<"String:"<<input<<" does not match prototype:"<<a<<std::endl;
    }
    return 0;
}
```

Έλεγχος αν ένα αλφαριθμητικό περιέχει 6 φορές το α