Baştan Sona C

Bir dilin kendine özgü kurallar bütününe “syntax” denir.

Programlamada alfanümerik olan üzerinde herhangi bir aritmetik işlem yapılmayacak sadece bilgi verme amacıyla yazılmış olan komutların yazdırılma işlemleri çift tırnak(“ ”) içerisinde yapılır.

Değişken, programda kullanacağımız verileri bellekte tutmamız için gerekn komutlardır.

“#include” komutu içe aktar anlamına gelir. Kütüphane eklemek için kullanırız.

“printf” komutu ekrana yazdır anlamına gelir.

“return 0” programı bitir anlamına gelir.

“//” bu karakter bize yorum satırı açmamızı sağlar.

Daha büyük yorum satırı için de “/\* \*/” karakterlerini gireriz ve arasına istediğimizi yazarız.

String= Harf dizinleri ve metinler için kullanılan değişken türüdür.

İnt = Tamsayı türündeki verileri saklamak için kullanılır.

Float = Kesirli sayıları saklamak için kullanılır.

Double = Kesirli sayıları saklamak için kullanılır. (Daha büyük)

Char = Karakter saklamak için kullanılır.

Her kodun bitimine noktalı virgül koyarız.

Metinleri ekrana yazdırmak istiyorsak “%s” kullanırız.

Eğer bir tam sayıyı ekrana yazdırmak istiyosak “%d” parametresini kullanırız.

Kesirli sayıları ekrana yazdırırken “%f” parametresini kullanırız.

Eğer daha büyük kesirli sayıları kullanacaksak “%lf” parametresini kullanırız.

Karakteri ekrana yazdırmak “%c” kullanırız.

Bir satırı alt satırı atmak istiyorsak alt satıra atmak istediğimiz cümlenin başına “\n” yazarız.

“Scanf” fonksiyonu kullanıcıdan veri almaya yarar.

Metinler için scanf kullanırken boşluk bırakmamaya dikkat etmeliyiz. Çünkü scanf boşluktan sonraki kelimeleri algılamaz.

Tanımlama yaparken” int isim = değer” yerine “#default isim değer” yazabiliriz.

C dilinde mod “%” ifadesi ile bulunur.

Bir değeri arttırmak veya azaltmak için arkasına işareti koyarsak(s++) ondan sonraki değerden itibaren arttırır, eğer önüne koyarsak(++s) o satır ve ondan sonraki değerler için geçerlidir.

OPRERATÖRLERİN ÖNCELİK SIRALAMALARI

OPERATÖR ÇALIŞMA YÖNÜ

() Soldan sağa

\*,/,% Soldan Sağa

+,- Soldan sağa

<,>,<=,>= Soldan sağa

==,!= Soldan sağa

= Sağdan sola

Programa rastgele bir değer ata demek için “int randomSayi=1+rand()%5” parmaetresini kullanırız. Yukarıda yazdığımız parametrede programa 1 ile 5 arasında rastgele sayı seç dedik

Programın seçtiği sayının sürekli değişmesini istiyorsak “srand(time(NULL));” satırını kullanırız.

Eğer kullanıcıdan alınan değer ile kendin tanımladığın değeri aynı olup olmadığını kontrol etmek istiyorsak “strcmp(tanımladığımızİsim, kullanıcıdanAlınanİsim)” şeklinde kullanabiliriz. Bunun içinde programımıza “string.h” kütüphanesini tanımlamamız gerekir.

C programlama dilinde 1’in 10’a bölümünden kalan 1 , 10’un 1’e bölümünden kalan ise 0’dır. Bu C programlama dilinde bir kuraldır.

Do-While döngüsünde while döngüsünden farklı olarak ilk başta komutu çalıştırır, ondan sonra koşula bakar.

“\t” komutu yana doğru bir tık boşluk bırakır.

For döngüsünde “for(koşullar){ çalıştırılacak kodlar}” şeklinde yazılır.

Break komutu verdiğinizden koşuldan itibaren çalışmayı durdurur(ayrıca koşuluda yazmaz).

Continue komutu, durdurduğunuz koşuldan itibaren çalışmay devam eder. Ama durdurduğumuz komutu çalıştırmaz.

Diziler kullanıcıdan birden fazla değer almamız gerektiğinde onları bir arada tutarak bize yardımcı olur.

Dizilerde tanımlama şöyle yapılır: ”int sayilar[]={“girdiğimiz sayilar”}

Bir dizideki elemanı ekrana yazdırmak için:”printf(“%d”, sayilar[istediğimiz elemanın indeksi]) kodu yazarız.

Dizilerde indeksler “0”dan başlar.

“strcat” komutu bir stringi diğer stringin sonuna ekler.”printf(strcat(isim,soyisim));” şeklinde kullanılır.

“strlen” komutu bir stringin kaç karakterden oluştığunu gösterir.”int sayi=strlen(isim) , printf(“%d”, sayi) şeklinde kullanılır.

“strcmp” komutu iki stringin aynı olup olmadığını kontrol eder.”strcmp(isim,soyisim)” şeklinde kullanılır.

“strlwr” komutu dizeyi küçük harfe çevirir.”printf(strlwr(isim));” şeklinde kullanılır.

“strupr” komutu dizeyi büyük harfe çevirir.”printf(strupr(isim));” şeklinde kullanılır.

“strrev” komutu dizeyi ters çevirir.”printf(strrev(isim));” şeklinde kullanılır.

Matrisler şöyle tanımlanır: “int matris[2][2]={{1,2},{3,4}}” . Matrisler de ilk köşeli parantezler matrisin boyutunu ifade eder. Bir de matrislerde sayılar sıfırdan başlar. Yani “1” in yerini tarif ederken “matris=[0][0]” anlarız.

Fonksiyonların görevleri, bir şeyi bir kez yazıp birden çok kez çağırabilmemizdir.

Örneğin; “void selamlamaMesaji(){“Birden fazla kez çağırmak istediğimiz mesaj”}” sonrada mesajı “main(){}” fonksiyonunun içinde çağırırız.

Fonksiyonlarda bir fonksiyondan diğer fonksiyona bilgi aktarımı yapmak için “return komutun adı” şeklinde gönderdiğimiz fonksiyonun içine yazarız.

Fonksiyonlarda “return 0” “false” ve “return 1” “true” sonucunu döndürür.

Doğru veya yanlışın değişkeni “bool” dur.

Bir sayının bellekteki adresini hektadecimal cinsten görmek için; “%x” parametresini yazarız. Ve virgülden sonraki kısmada “&” ile değişkenin adını yazarız.

Bir programa pointer tanımlarken pointerın başına “\*” işaretini koyarız. Bu hem bilgisayar hem de bize değişkenin pointer olduğunu gösterir.

Pointer in tanımı; Bellekle program arasında bağlantı kurma işlemlerinde kullandığımız yapıdır.

Pointerlar herhangi bir değişkenin kendisiyle yada adresiyle işlem yapmamızı sağlar. Pointerın adresini bulma örnek kullanım:

“printf(“%x” , &tanımladığınız değişkenin adı);”

Pointerın adresini ekrana yazdırmak için ikinci bir kullanım:

“printf(“Adres:%x” , tanımladığınız pointerın adı);”

C dilinde bir değişkenin pointer olduğunu göstermek için başına “\*” işareti koyulur.

Pointerlarda dizi kullanmak için & operatörüyle birlikte dizinin ismini pointer a atarız.

Yapılar, farklı türden değişkenleri bir arada tutan değişkendir. Örneğin; “struct” gibi.

Typedefin özellikleri:

1-) Typedef kullanınca yapımız için yeni değişken tanımlarken struct kullanmak zorunda kalmayız.

2-) İstediğiniz türde değişkeni bir kez tanımlayıp, sonra onu takma ad olarak kullanarak bir daha aynı türde değişkeni tanımlayarak zorunda kalmıyorsunuz.

ÖRNEK KULLANIM:

typedef int tamsayi;

int main(){

tamsayi ts;

ts=22;

printf(“%d”,ts);

return 0;

}

Enum , dallanmaların çok olduğu , içerisindeki dizinin elemanlarının fazla olduğu, birbirleriyle ilişkili verilerde kullanılan yapılardır.

ÖRNEK KULLANIM:

enum sehirler{

adana,adiyaman,afyon,agri,amasya,ankara,Antalya

};

İnt main(){

Enum sehirler il;

İl= amasya;

Printf(“%d”,il);

Burada dizideki amasya elemanının indeksini bulduk.