**TEMEL KAVRAMLAR**

**Merkezi İşlem Birimi:**

Bilgisayardaki tüm karar verme ve kontrol işlemlerini gerçekleştirir. Matematiksel işlemleri gerçekleştirdiği gibi bilgisayarda hangi birimlerden giriş yapılacak hangi birimlerden çıkış yapılacak öncelik sırasına göre işlemi gerçekleştirir.

**Bellek Birimi:**

Bilgisayarlar çalıştıkları süre boyunca giriş biriminden aldığı veya hesaplama sonucu elde ettiği verileri bellek üzerinde saklayarak işlemleri gerçekleştirir.

**Giriş/Çıkış Birimleri:**

Kullanıcıdan veya diğer aygıtlardan (Fare, Klavye, Mikrofon kamera, Tarayıcı vb.) bilgisayara veri aktarmak için kullanılan birimlere Giriş Birimleri; bilgisayarda bulunan verileri kullanıcıları bilgilendirmek amacıyla veya diğer aygıtlara (ekran, yazıcı, tarayıcı, hoparlör, kulaklık, vb.) göndermek amacıyla kullanılan birimlere çıkış birimleri diyoruz.

**Yazılım:**

Bilgisayarın çalışması için donanım dışında kalan kısma yazılım denilir. Yani yapılması gereken işleri yapabilmek için donanıma komutlar veren programlar topluluğudur.

Genel olarak üç kısımda incelenir:

Sistem Yazılımları (İşletim Sistemi – Windows, Unix, Linux, Debian, vb.)

Program Geliştirme Yazılımları (Programlama Dilleri – Java, C#, C++, C, Python, vb.)

Uygulama Yazılımları (MS Office, Autocad, Photoshop,3D MAX)

Yazılım geliştirme sonucu ortaya çıkan ürüne program diyoruz.

Bir problemin bilgisayar tarafından çözülebilmesi için öncelikle algoritmasının oluşturulması gerekiyor.

**Matematiksel İşlemler**

|  |  |  |
| --- | --- | --- |
| İşlem | Matematik | Bilgisayar |
| Toplama | **A+B** | **A+B** |
| Çıkarma | **A-B** | **A-B** |
| Çarpma | **AXB || A . B** | **A\*B** |
| Bölme | **A : B** | **A/B** |
| Üs Alma | **Ab** | **A^B** |

Matematiksel işlemlerin öncelik sırası?

|  |  |  |
| --- | --- | --- |
| Sıra | İşlem | Bilgisayar Dili |
| 1 | Parantezler | (…(……….)…) |
| 2 | Üs Alma | A^B |
| 3 | Çarpma Bölme | A\*B || A/B |
| 4 | Toplama Çıkarma | A+B || A-B |

NOT: Bilgisayar diline kodlanmış bir matematiksel ifade, aynı önceliğe sahip işlemler mevcut ise bilgisayarın bu işlemleri gerçekleştirme sırası soldan sağa (Baştan Sona) doğrudur.

Örneğin Y=A\*B/C

Karşılaştırma Karar İfadeleri

İki büyükten hangisinin büyük veya küçük olduğu

İki değişkenin birbirine eşit olup olmadığı gibi konulara karar verir.

|  |  |
| --- | --- |
| İşlem Sembolü | Anlamı |
| = | Eşittir. |
| != || <> | Eşit Değildir. |
| > | Büyüktür. |
| < | Küçüktür. |
| >= veya => | Büyük Eşittir. |
| <= veya =< | Küçük Eşittir. |

**Algoritma Kavramı**

Algoritma, en basit ifadeyle, bir problemi çözmek için takip edilecek sonlu sayıda adımdan oluşan bir çözü yoludur.

Diğer bir ifadeyle algoritma, bir problemin mantıksal çözümünün adım adım nasıl gerçekleştireceğininin sözlü ve kağıda dökülmüş halidir.

* Algoritma ile oluşturulan sözel olarak ifade edildiğinden daha standart herkesin gördüğünde ortak olarak aynı sonucu çıkarabileceği hale getirmek için akış diyagramları kullanılır. Akış Diyagramlarının sembollerden oluştuğunu unutmadan Her sembolün bir işlevi vardır.
* Algoritması oluşturulmuş bir problemin bilgisayar ortamına aktarılış haline program denir.

İki Sayının Toplamını yapan algoritmayı oluştur.

Başla

A ve B değişkenlerini bilgisayara tanıttım

A Sayısının Değerini Girdik

B Sayısının Değerini Girdik

A ve B Sayısını Topladık

Çıkan Sonucu Ekrana Yazdırdım

Bitiş.

Problem Çözmek Adımları:

Problem çözmede iki temel yöntem vardır.

Deneme Yanılma Yöntemi

Algoritma geliştirmek

1. Problemi Tanımlamak: Algoritmanın amacı belirli bir problemi çözmek ise algoritma geliştirmenin esas öğesi problem kendisi. Problemi iyi anlamamız gerekiyor. Problemi iyi anlamazsak bize istediğimiz sonucun bambaşka bir versiyonunu ortaya koyar bu bizi batırır.
2. Girdi ve Çıktıları Belirlemek: Problemi iyi tanımlama için başlangıç ve bitiş noktalarını çok net belirlememiz gerekiyor. Bizim bulacağımız şey problemin aslında çözümüdür. Bizim elimizde parametre yoksa zaten bu iş olmaz.
3. Çözüm Yolları Geliştirme: Bir problemin çözümü için çoğunlukla birden fazla seçeneğimiz olması gerekiyor. İçinde bulunduğumuz duruma göre bazen zaman sıkışıklığından ilk bulduğumuz çözüm yolunu uygulamak zorunda kalırız.
4. Çözüm Sınaması ve İyileştirilmesi: Algoritmayı geliştirdikten sonra henüz kodlamadan kağıt üzerinde nasıl çalışacağını sınamalıyız. Bunu Yaptığımızda eğer algoritmada bir eksiklik ve hata çıkıyorsa bunu düzeltmeli ve tekrar sınamalıyız.

**Örnekler**

Kullanıcının girdiği iki sayının karelerinin toplamını görüntüleyen programın algoritmasını yazınız.

Başla

A, b, toplama

A sayısını Biriniz

B Sayısını Giriniz

Toplama=(A2) + (B2)

Toplamayı ekrana yazdır.

Bitir

Doğum Tarihi girilen kişinin yaşını hesaplayan programın algoritmasını yazın.

Başla

Dogumtarihi, AnlikYili, Yas

Doğum Tarihini Gir

Anlık Yılı Gir

Yas= AnlikYili-Dogumtarihi

Ekrana yasi yazdır.

Bitir

Girilen sayının faktöryelini bulan programın algoritmasını yazınız.

Başla

Sayi, Fak=1

Sayı Giricez

Fak=Fak\*Sayi

sayi - -

Eğer Sayi=1 devam et

Değilse 4’e git diyeceğiz

Yazdır Faktöryeli

Bitir

Girilen sayının Çift veya Tek olduğunu bulan programın algoritmasını yazınız.

Başla

Sayi1

Sayı değerini girdik

Sayi %2 = 0

Sayı Ekrana Çift Gelir Yazdır.

Sayi %2=1

Sayı Ekrana Tek Gelir Yazdır

Bitir

Girilen Sayının Negatif ve Pozitif Olan programın algoritmasını yazın.

Başla

Sayı

Sayı değer alınız

Sayı>0

Ekrana Pozitif Yazdır

Sayı<0

Ekrana Negatif Yazdır.

Sayi=0

Ekrana Nötr Yazdır.

Bitir.

Girilen Sayının Kaç Basamaklı olduğunu hesaplayan programın algoritmasını yazınız.

Başla

Sayı, sayac=1

Sayının değerini istedik

Eğer sayı<=9 ise 6’ya git

Değilse devam et

Sayı/10

sayac++

Yazdır Sayac

Bitir