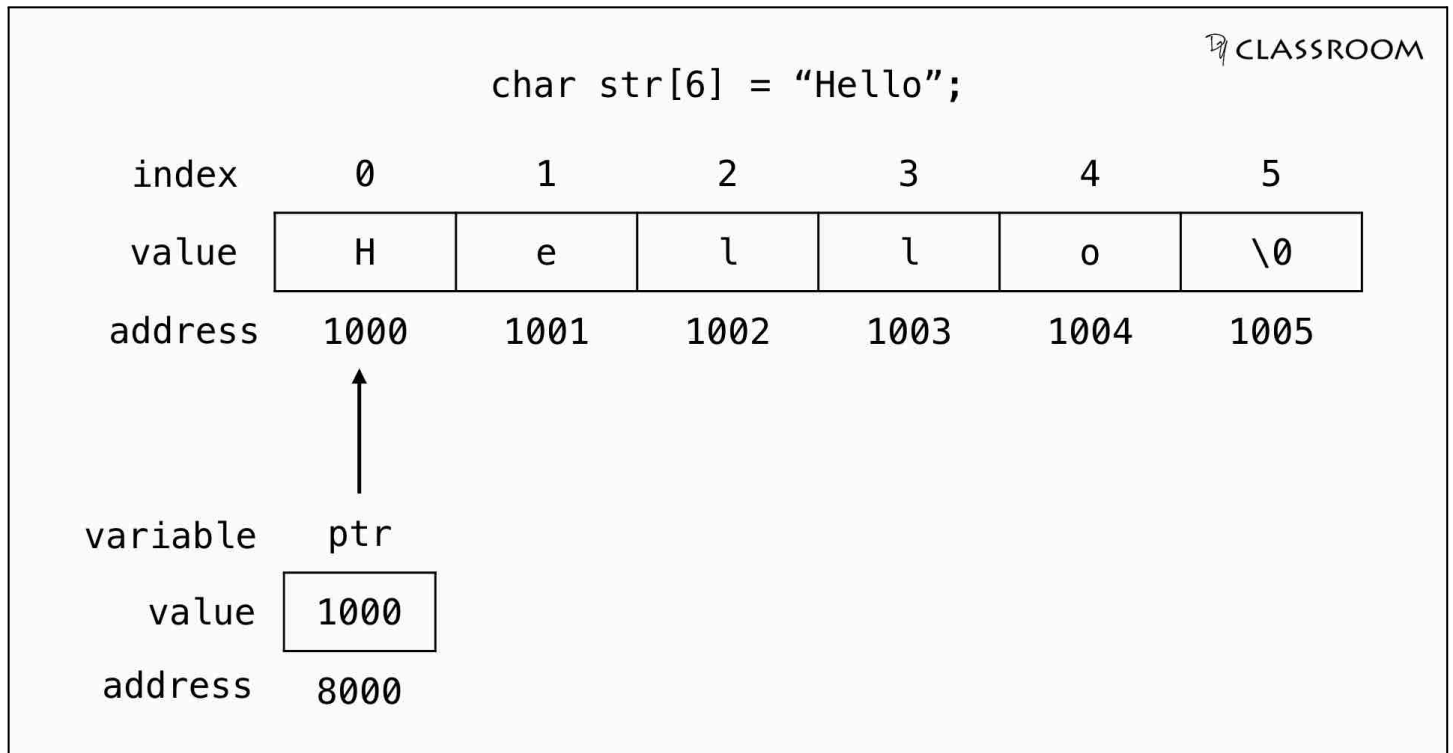


Pointer

Pointer Nedir?

Pointer ve String Kullanımı

<https://dyclassroom.com/c/c-pointers-and-strings>



```
#include <stdio.h>
```

```
int main(void) {
```

```
    // string variable  
    char str[6] = "Hello";
```

```
    // pointer variable  
    char *ptr = str;
```

```
    // print the string  
    while(*ptr != '\0') {  
        printf("%c", *ptr);
```

```
        // move the ptr pointer to the next memory location  
        ptr++;  
    }
```

```
    return 0;  
}
```

Dersteki örnek

```
#include <stdio.h>

int main(void) {

    // string variable
    char kursiyer[11] = "UmutEgeAta";    // str="UmutEgeAta" !! python x C++

    // pointer variable
    char *ptr  = kursiyer;
    char *ptr2 = kursiyer;

    // adres ve adresin göst. degeri
    while(*ptr != '\0') {
        printf("\nAdres: %p", ptr);
        printf("\nDeger: %c", *ptr);
        ptr++;
    }

    // print the string
    printf("\n\n");
    while(*ptr2 != '\0') {

        printf("%c", *ptr2);

        // move the ptr pointer to the next memory location
        ptr2++;
    }

    return 0;
}
```

Ödev: String de rakam kontrolü

Pointer ve diziler

Kaynak:

https://www.bilgigunlugum.net/prog/cprog/c_switch

<https://www.cagataycebi.com/>

C dilinde pointer,

bir değişkenin bellekteki adresini tutan bir değişkendir. Adresi tuttuğu değişkenin değerine erişmek için kullanılır.

Örnek:

```
#include <stdio.h>
```

```

int main() {
    int num = 10;
    int *p;

    p = &num; // 'p' pointer değişkenine 'num' değişkeninin bellek adresini atayın.

    printf("num değişkeninin değeri: %d\n", num);
    printf("num değişkeninin bellek adresi: %p\n", &num);
    printf("p değişkeninin bellek adresi: %p\n", p);
    printf("p değişkeninin gösterdiği değer: %d\n", *p);

    return 0;
}

```

Bu örnekte, num adında bir int değişkeni tanımladık ve p adında bir int pointer değişkeni tanımladık. Daha sonra, & operatörünü kullanarak num değişkeninin bellek adresini p değişkenine atadık.

Sonra, printf() fonksiyonunu kullanarak num değişkeninin değeri, bellek adresi, p değişkeninin bellek adresi ve p değişkeninin gösterdiği değerleri yazdırdık. * işaretçi operatörü, bir pointer değişkenin gösterdiği adresin içeriğine (değer) erişmek için kullanılır.

Bu programı çalıştırdığınızda, num değişkeninin değeri, bellek adresi, p değişkeninin bellek adresi ve p değişkeninin gösterdiği değer (yani num değişkeninin değeri) yazdırılır.

Fonksiyon parametresi olarak pointer göndermek ile ilgili örnek

```

#include <stdio.h>

void degeriDegistir(int *p) {
    *p = 100; // pointerin gösterdiği bellek adresindeki değeri 100 olarak değiştir
}

int main() {
    int sayi = 10;
    printf("Fonksiyondan önce sayi değişkeninin değeri: %d\n", sayi);

    degeriDegistir(&sayi); // sayi değişkeninin adresini fonksiyona gönder
    printf("Fonksiyondan sonra sayi değişkeninin değeri: %d\n", sayi);

    return 0;
}

```

Bu örnekte, degeriDegistir adında bir fonksiyon tanımladık. Fonksiyon, int tipinde bir pointer (*p) parametresi alır. Fonksiyon içinde, *p işaretçi operatörü kullanılarak, pointerin gösterdiği bellek adresindeki değeri 100 olarak değiştirir.

Daha sonra, main fonksiyonunda sayi adında bir int değişkeni tanımlarız. printf() fonksiyonunu kullanarak, sayi değişkeninin başlangıç değerini yazdırırız.

Sonra, `degeriDegistir` fonksiyonunu `&sayi` ifadesi ile çağırarak, `sayi` değişkeninin bellek adresini fonksiyona göndeririz. Bu, `degeriDegistir` fonksiyonunun, `sayi` değişkeninin değerini değiştirebilmesini sağlar.

Son olarak, `main` fonksiyonunda, `printf()` fonksiyonunu kullanarak `sayi` değişkeninin değerinin değişip değişmediğini kontrol ederiz. Bu örnekte, `sayi` değişkeninin değeri 100 olacaktır, çünkü `degeriDegistir` fonksiyonu `sayi` değişkeninin adresini alır ve değeri 100 olarak değiştirir.

Pointer ve dizi kullanımı örneği

```
#include <stdio.h>

int main() {
    int dizi[] = {10, 20, 30, 40, 50}; // 5 elemanlı bir dizi tanımla
    int *p;

    p = &dizi[0]; // 'p' pointer değişkenine 'dizi' dizisinin ilk elemanının bellek adresini ata

    for(int i=0; i<5; i++) {
        printf("dizi[%d] = %d\n", i, *(p+i)); // p pointer değişkeninin i. elemanının içeriğini yazdır
    }

    return 0;
}
```

Bu örnekte, `dizi` adında bir `int` dizisi tanımladık. Daha sonra, `int` tipinde bir pointer (`*p`) tanımlarız. `p` değişkenine, `&dizi[0]` ifadesini kullanarak `dizi` dizisinin ilk elemanının bellek adresini atadık. Bu, `p` değişkeninin dizinin ilk elemanını işaret etmesini sağlar.

Sonra, bir `for` döngüsü kullanarak, `p+i` ifadesi ile `p` pointer değişkeninin `i` nci elemanının bellek adresine erişiriz. `*(p+i)` ifadesi ile de, pointerin gösterdiği bellek adresindeki değeri (yani `dizi[i]` nin değerini) yazdırırız.

Bu programı çalıştırdığınızda, `dizi` dizisindeki elemanların değerleri, `p` pointer değişkeni kullanılarak yazdırılır. `p` pointer değişkeni, `dizi` dizisinin ilk elemanını gösterdiği için, `*(p+i)` ifadesi ile `dizi[i]` nin değerleri yazdırılır.

String veriler ile pointer kullanımı örneği

```
#include <stdio.h>

int main() {
    char *mesaj = { "Merhaba, dünya!" }; // char türünden bir pointer kullanarak string
    veri tanımlama

    printf("Mesaj: %s\n", mesaj); // pointer kullanarak string veriyi yazdırma

    return 0;
}
```

Bu örnekte, `char` türünden bir pointer (`*mesaj`) tanımladık ve `mesaj` değişkenine, "Merhaba, dünya!" gibi bir string veri atadık. Bu, bir string verisinin bellek adresini gösteren bir pointer değişkenidir.

Daha sonra, printf() fonksiyonunu kullanarak mesaj değişkeninin içeriğini yazdırırız. %s format belirleyicisini kullanarak, mesaj pointer değişkeninin gösterdiği bellek adresindeki string veriyi yazdırırız.

Bu programı çalıştırdığınızda, "Merhaba, dünya!" gibi bir string verisi çıktı olarak yazdırılacaktır.

```
#include <stdio.h>

int main() {
    char *aylar[] = { "OCAK", "SUBAT", "MART" }; // char türünden bir pointer kullanarak string veri tanımlama
    int i;

    printf("Mesaj: %s\n\n", *aylar); // pointer kullanarak string veriyi yazdırma
    for ( i=0; i<3; i++)
        printf("%s\n", aylar[i]);

    return 0;
}
```

C dilinde string dizinin eleman sayısı

C dilinde, bir string dizisinin eleman sayısını belirlemek için strlen() fonksiyonu kullanılabilir. Bu fonksiyon, bir string verisinin uzunluğunu hesaplar ve geri döndürür. String uzunluğu, dizinin eleman sayısına eşittir.

Aşağıdaki örnekte, bir string dizisinin eleman sayısını strlen() fonksiyonu kullanarak hesaplayabilirsiniz:

```
#include <stdio.h>
#include <string.h>

int main() {
    char dizi[] = "Merhaba, dünya!"; // string dizisi tanımlama
    int eleman_sayisi;

    eleman_sayisi = strlen(dizi); // strlen() fonksiyonu kullanarak eleman sayısını hesapla

    printf("Dizinin eleman sayısı: %d\n", eleman_sayisi); // eleman sayısını yazdır

    return 0;
}
```

Bu örnekte, dizi adında bir string dizisi tanımladık. eleman_sayisi adında bir tamsayı değişkeni de tanımladık. Daha sonra, strlen() fonksiyonunu kullanarak dizi dizisinin eleman sayısını hesapladık ve eleman_sayisi değişkenine atadık.

Son olarak, printf() fonksiyonunu kullanarak eleman_sayisi değişkeninin içeriğini yazdırdık.

Bu programı çalıştırdığınızda, "Dizinin eleman sayısı: 16" gibi bir çıktı elde edeceksiniz. Bu, dizi dizisinin 16 elemandan oluştuğunu gösterir.

Haftanın günlerini gösteren bir string dizisi tanımlar

Aşağıdaki örnek, haftanın günlerini gösteren bir string dizisi tanımlar ve haftanın üçüncü gününü yazdırır:

```
#include <stdio.h>

int main() {
    char *haftanın_gunleri[] = {"Pazartesi", "Sali", "Carsamba", "Persembe", "Cuma", "Cumartesi", "Pazar"};
    int index = 2; // haftanın üçüncü günü "Carsamba" olduğu için 2. indeksi seçiyoruz

    printf("Haftanın ucuncu gunu: %s\n", haftanın_gunleri[index]);

    return 0;
}

//----- 0. değeri değiştiriyoruz.
#include <stdio.h>

int main() {
    char *haftanın_gunleri[] = {"Pazartesi", "Sali", "Carsamba", "Persembe", "Cuma", "Cumartesi", "Pazar"};
    int i = 0;

    for ( i=0; i<7; i++)
        printf("Haftanın %d. gunu: %s\n", i+1, haftanın_gunleri[i]);

    haftanın_gunleri[0] = "yeniiii degeeeerrr";
    printf("\nDegisen deger: %s\n", haftanın_gunleri[0]);

    return 0;
}
```

Bu örnekte, `haftanın_gunleri` adında bir string dizisi tanımladık ve içinde haftanın günlerinin adları bulunuyor. `index` adında bir tamsayı değişkeni tanımladık ve bu değişkene "Carsamba" gününün indeks numarası olan 2 değerini atadık.

Daha sonra, `printf()` fonksiyonunu kullanarak `haftanın_gunleri[index]` ifadesini yazdırdık. Bu ifade, `haftanın_gunleri` dizisinin 2. elemanını (yani "Carsamba" gününü) temsil eder.

Bu programı çalıştırdığınızda, "Haftanın ucuncu gunu: Carsamba" gibi bir çıktı elde edeceksiniz.

C'de bir dizinin boyutunu veren kodu hesaplamak için,

dizinin toplam eleman sayısını, elemanların boyutuna bölerek boyutunu hesaplayabiliriz. Bunu yapmak için, aşağıdaki örnek kodu kullanabilirsiniz:

```
#include <stdio.h>

int main() {
    int dizi[] = {1, 2, 3, 4, 5};
    int boyut = sizeof(dizi) / sizeof(dizi[0]); // veya
```

```
// int boyut = sizeof(dizi) / sizeof(int); dizinin tipi int

printf("Dizinin boyutu: %d\n", boyut);

return 0;
}
```

Bu program, bir tamsayı dizisi tanımlar ve boyutunu hesaplar. Boyut hesaplanırken, sizeof operatörü kullanılır ve dizinin toplam byte cinsinden boyutunu döndürür. Böylece, elemanların boyutunu `sizeof(dizi[0])` kullanarak hesaplarız ve toplam eleman sayısını bu değere bölerek, dizinin eleman sayısını elde ederiz.

```
#include <stdio.h>

int main(void)
{
    char cdizi[] = { 'A', 'B', 'C', 'D', 'E', 'F' };
    char *cp;
    int len, id;

    cp = cdizi; // cp = &cdizi[0];
    len = sizeof(cdizi) / sizeof(char);

    for (id=0; id<len; id++) printf("%c ", *cp++);

    return 0;
}
```

Program,

- cdizi adlı bir diziye ilk değer atama yöntemi ile A-F arasındaki harfleri atar.
- Sonra, cp adlı bir işaretçi ile len ve id adlı iki adet int değişken bildirimi yapar.
- cdizi dizisinin bellek adresini cp işaretçisine atar.
- Dizi eleman sayısını, dizi boyutunu char veri türü boyutuna bölerek elde eder ve len değişkenine atar.
- Bir for döngüsü, dizi eleman değerlerini ekrana yazar.
- for döngüsünün her tekrarında **cp işaretçi değeri bir byte (char veri türü için) kadar artırılarak**, bir sonraki dizi elemanının bellek adresini göstermesi sağlanır.

Pointer üzerinde işlemler

```
#include <stdio.h>

int main(void)
{
    int *ip;
    int idizi[5] = { 5, 17, 21, 34, 46 };
    int id;

    printf("İşaretçi bellek adresi: %p\n\n", &ip);

    ip = idizi; // ip = &idizi[0];
```

```
for (id=0; id<5; id++, ip++) {  
    printf("İşaretçinin içerdiği bellek adresi: %p %p\n", ip, &idizi[id]);  
    printf("İşaretçinin gösterdiği değişken değeri: %d %d\n\n", *ip, idizi[id]);  
}  
  
return 0;  
}
```

Yukarıdaki programı derleyip çalıştırdığımızda, aşağıdaki ifadeleri ekrana yazar:

İşaretçi bellek adresi: 006dfed8

İşaretçinin içerdiği bellek adresi: 006dfec4 006dfec4

İşaretçinin gösterdiği değişken değeri: 5 5

İşaretçinin içerdiği bellek adresi: 006dfec8 006dfec8

İşaretçinin gösterdiği değişken değeri: 17 17

İşaretçinin içerdiği bellek adresi: 006dfec4 006dfec4

İşaretçinin gösterdiği değişken değeri: 21 21

İşaretçinin içerdiği bellek adresi: 006dfed0 006dfed0

İşaretçinin gösterdiği değişken değeri: 34 34

İşaretçinin içerdiği bellek adresi: 006dfed4 006dfed4

İşaretçinin gösterdiği değişken değeri: 46 46

Program, önce ip adlı int bir işaretçi ile idizi adlı 5 elemanlı int bir dizi tanımlar. İşaretçi bellek adresini ekrana yazdıktan sonra, dizinin ilk elemanının adresini işaretçiye atar. Bir for döngüsü içinde, her defasında işaretçi değerini int boyutu kadar artırmak suretiyle, işaretçinin bir sonraki dizi elemanının adresini göstermesini sağlayarak işaretçinin içerdiği adresi ve gösterdiği dizi değişken değerini hem işaretçi adı hem de dizi adını kullanarak ikişer defa ekrana yazar.

for döngüsüne ilk girdiğinde bellek durumu aşağıdaki ilk şekilde gibidir. Döngü içinde ip++ ifadesi ilk çalıştığında ise, ip int bir işaretçi olduğundan, int boyutu kadar (32 bit bilgisayarlarda 4 byte) yukarıdaki bellek adresinin değerini alır. Yani, daha önce idizi[0] değişken adresini içeren ip işaretçisi artık idizi[1] değişkeninin adresini içermektedir. Aslında, işlem satırındaki ip++ ifadesinin gerçek değeri aşağıdaki gibidir.

```
ip = ip + (1*sizeof(int));
```


İşlem satırı		int idizi[5] = { 5, 17, 21, 34, 46 };	ip = idizi;	ip++;	ip++;	ip++;	ip++;
Değişkenler	Bellek adresi	Adres içeriği					
idizi[0]	0x6dfec4	5	5	5	5	5	5
idizi[1]	0x6dfec8	17	17	17	17	17	17
idizi[2]	0x6dfecc	21	21	21	21	21	21
idizi[3]	0x6dfed0	34	34	34	34	34	34
idizi[4]	0x6dfed4	46	46	46	46	46	46
*ip	0x6dfed8		0x6dfec4	0x6dfec8	0x6dfecc	0x6dfed0	0x6dfed4

Sonuç olarak, for döngüsünün her tekrarında ip++ işlemi ile ip işaretçisinin içerdiği adres int boyutu kadar artarak bir sonraki dizi değerinin adresini gösterir.

ÖNEMLİ BİLGİ:

* Bir işaretçiye eklenen veya çıkarılan değer mutlaka int bir değer olmalıdır.

* İşaretçilerle toplama veya çıkarma dışında bir işlem yapılamaz.

Web kaynak: https://www.bilgigunlugum.net/prog/cprog/c_isaretci#c_isa01

İşaretçiler yoluyla değişken değerlerinin değiştirilmesi

++ ve -- operatörleri ile normal toplama ve çıkarma işlemlerini kullanarak işaretçinin adresini gösterdiği değişken değerini dolaylı yöntemle artırabiliriz:

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    int *ip, id;
```

```
    id = 274;
```

```
    ip = &id;
```

```
    printf("id değişken değeri: %d\n", id);
```

```
    // ip işaretçisinin adresini gösterdiği değişken değerini artırma
```

```
    (*ip)++;
```

```
    printf("id değişken değeri: %d", id);
```

```
    return 0;
```

```
}
```

Yukarıdaki programı derleyip çalıştırdığımızda, aşağıdaki ifadeleri ekrana yazar:

id değişken değeri: 274

id değişken değeri: 275

Program, ip adlı int bir işaretçi ve id adlı int bir değişken oluşturur. id değişkenine 274 sayısını atar. id değişkeninin adresini ip işaretçisine atar. Değişken değerini ekrana yazdıktan sonra ip işaretçisini kullanarak, id değişken değerini artırır ve değişken değerini tekrar ekrana yazar.

Bir pointer kullanarak bir değişkenin değerini arttırma örneği verilmiştir:

```
#include <stdio.h>

int main() {
    int sayi = 10;
    int *ptr;

    ptr = &sayi;
    (*ptr)++;

    printf("Sayi: %d\n", sayi);

    return 0;
}
```

Bu program, bir tamsayı değişkeni tanımlar ve bir pointer tanımlar. Pointer, değişkenin adresine atanır. Daha sonra, **(*ptr)++** ifadesi kullanılarak, pointer aracılığıyla değişkenin değeri 1 artırılır.

Bu çıktı, değişkenin değerinin 1 arttırıldığını gösterir. Pointer'lar, özellikle fonksiyonlara parametre olarak kullanıldığında, değişkenleri doğrudan değiştirmek için kullanışlıdır.

Klavyeden girilen 3 kelimeden ilk harfi A ile başlayanları listeleyen program

Aşağıdaki örnek program, klavyeden girilen üç kelime içinde ilk harfi "A" ile başlayan kelimeleri listeleyen bir programdır ve dizi ve pointer kullanarak yazılmıştır:

```
#include <stdio.h>
#include <string.h>

#define MAX_KELIME_UZUNLUGU 50

int main() {
    char kelimeler[3][MAX_KELIME_UZUNLUGU];
    char *ptr[3];

    printf("Lutfen uc kelime giriniz:\n");
    scanf("%s %s %s", kelimeler[0], kelimeler[1], kelimeler[2]);
```

```

ptr[0] = kelimeler[0];
ptr[1] = kelimeler[1];
ptr[2] = kelimeler[2];

for (int i = 0; i < 3; i++) {
    if (ptr[i][0] == 'A') {
        printf("%s\n", ptr[i]);
    }
}

return 0;
}

```

Bu program, üç kelimeyi "kelimeler" adlı bir diziye yerleştirir ve her kelime için bir pointer tanımlar. Pointers, dizi elemanlarının adreslerine atanır.

Daha sonra, bir döngü içinde her bir kelimenin ilk harfinin "A" ile başlayıp başlamadığı kontrol edilir. Eğer bir kelime "A" ile başlıyorsa, printf() fonksiyonu kullanılarak kelime ekrana yazdırılır.

Lutfen uc kelime giriniz:

Adana Ankara Antalya

Adana

Ankara

Klavyeden girilen kelimeleri ilk harfine göre ayrı dizilere yerleştiren bir örnektir:

```

#include <stdio.h>
#include <string.h>

#define MAX_KELIME_UZUNLUGU 50
#define MAX_KELIME_SAYISI 10

int main() {
    char kelimeler[MAX_KELIME_SAYISI][MAX_KELIME_UZUNLUGU];
    char baslikHarfleri[26] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    int kelimeSayilari[26] = { 0 };
    int i, j;

    printf("Lutfen en fazla 10 adet kelime giriniz:\n");

    for (i = 0; i < MAX_KELIME_SAYISI; i++) {
        scanf("%s", kelimeler[i]);

        if (kelimeler[i][0] >= 'A' && kelimeler[i][0] <= 'Z') {

```

```

        kelimeSayilari[kelimeler[i][0] - 'A']++;
    }
}

for (i = 0; i < 26; i++) {
    if (kelimeSayilari[i] > 0) {
        printf("\nKelimeler baslayan harfi '%c' olan dizide:\n", baslikHarfleri[i]);

        for ( j = 0; j < MAX_KELIME_SAYISI; j++) {
            if (kelimeler[j][0] == baslikHarfleri[i]) {
                printf("%s\n", kelimeler[j]);
            }
        }
    }
}

return 0;
}

```

Bu program, klavyeden en fazla 10 kelime alır ve her kelimenin ilk harfini kontrol eder. Dizilerin birinci boyutu, kelimelerin sayısını, ikinci boyutu ise kelime uzunluğunu belirtir. Kelimeler, "kelimeler" adlı bir diziye yerleştirilir.

"baslikHarfleri" adlı bir karakter dizisi, 26 büyük harf içerir. "kelimeSayilari" adlı bir tamsayı dizisi, her harf için kelime sayısını saklar.

Daha sonra, bir döngü içinde her bir kelimenin ilk harfinin kontrol edilir. Eğer kelimenin ilk harfi büyük harfse, "kelimeSayilari" dizisinde ilgili harf için bir sayaç artırılır.

Son olarak, bir döngü içinde her harf için kelime sayısı kontrol edilir. Eğer bir harf için en az bir kelime varsa, o harfin dizisi oluşturulur ve ekrana yazdırılır.

Programı çalıştırdığımızda, örneğin aşağıdaki gibi bir çıktı elde edebiliriz:

Lutfen en fazla 10 adet kelime giriniz:

elma armut ayva ananas angora biber brokoli ceviz dere ekmek

Kelimeler baslayan harfi 'A' olan dizide:

ananas

Kelimeler baslayan harfi 'B' olan dizide:

biber

brokoli

Kelimeler baslayan harfi 'C' olan dizide:

ceviz

-----/ 2.defa çalıştırılım /-----

Lutfen en fazla 10 adet kelime giriniz:

ANANAS ELMA ARMUT INCIR AYVA HURMA PORTAKAL MANDALINA ERIK DOMATES

Kelimeler baslayan harfi 'A' olan dizide:

ANANAS

ARMUT

AYVA

Kelimeler baslayan harfi 'D' olan dizide:

DOMATES

Kelimeler baslayan harfi 'E' olan dizide:

ELMA

ERIK

Kelimeler baslayan harfi 'H' olan dizide:

HURMA

Kelimeler baslayan harfi 'I' olan dizide:

INCIR

Kelimeler baslayan harfi 'M' olan dizide:

MANDALINA

Kelimeler baslayan harfi 'P' olan dizide:

PORTAKAL