

İSKENDERUN TEKNİK ÜNİVERSİTESİ Mühendislik ve Doğa Bilimleri Fakültesi Bilgisayar Mühendisliği Bölümü BLM2-1112 Algoritmalar ve Programlama II Vize Sınavı		Toplam Puan:				
		Soru	S1 (30p)	S2 (20p)	S3 (20p)	S4 (30p)
Öğrenci No:	Tarih: 28.04.2025	ÖÇ	1,2,3	4	5	6
Ad-Soyad :	Süre:75 dk.	PÇ	3	4	4	5
İmza:	<input type="checkbox"/> 1. Örgün <input type="checkbox"/> 2. Örgün	Puan:				

S-1) Aşağıdaki test sorularında en uygun seçeneği işaretleyiniz. (Her test sorusu 3 puandır.)

T1. Aşağıda verilen kod parçacağında bulunan struct yapısı ile ilgili tanımlamalardan hangisi doğrudur?

```
struct Person {
    char name[50];
    int age;
};
```

- A) Person p = {"Peri", 25};
- B) struct Person p = {"Peri", 25};
- C) Person p = new Person;
- D) Person p;
- E) struct Person p;

T2. Aşağıdaki kodun çıktısı nedir?

```
#include<stdio.h>

int deger=0b0011^0b0011;

int main(){
    printf("%d", deger);
}
```

- A) 0000
- B) 1111
- C) 15
- D) 0
- E) 1

T3. Aşağıdaki program parçası ile ilgili hangi ifade doğrudur?

```
int main(){
    int n[50];
    for (size_t i = 0; i < 50; ++i) {
        n[i] = 15*i+2;
    }
}
```

- A) n dizisi için 50 birimlik bellek adresi tanımlanmıştır.
- B) i=5 için n[5]=78 olur.
- C) Bu dizinin başlangıç elemanı n[1] ile, son elemanı ise n[50] ile tanımlanmıştır.
- D) for döngüsü içerisinde "size_t" kullanılması yanlıştır. Burada mutlaka "int" kullanılması gerekirdi.
- E) Bu dizinin elemanları reel sayıdır.

T4. Aşağıda verilen program ile ilgili hangisi doğrudur?

```
#include <stdio.h>

#define RESP 20
#define FREQ 11

int main(void){
    int frequency[FREQ] = {0};
    int responses[RESP] = {1, 6, 6, 4, 6, 5, 9, 6, 6, 1, 1, 6, 3, 6, 6, 10, 3, 6, 10, 3};
    for (size_t ans = 0; ans < RESP; ++ans) {
        ++frequency[responses[ans]];
    }
    for (size_t rat = 1; rat < FREQ; ++rat) {
        printf("%6d \n", frequency[rat]);
    }
}
```

- A) 20 elemanlı “responses” dizisi ile 11 elemanlı “frequency” dizileri yazdırılır.
- B) RESP ve FREQ değişkenleri dizi değerlerini atamak için kullanılmıştır.
- C) “frequency” dizisinin tüm elemanlarının başlangıç değerlerine “0” atanmıştır.
- D) “responses” dizisine 19 tane sayısal değer girilmiştir. Son değer \0 (null) olur.
- E) “responses” dizisindeki sayısal değerlerin ortalaması “frequency[rat]” dizisindedir.

T5. Aşağıda verilen program çalıştırıldığında, array1 içerisindeki değerler aşağıdakilerden hangisidir?

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

```
int main(void){
    for(size_t i = 0; i<5; i++){
        static int array1[5];
        for (size_t j = 0; j < 5; ++j) {
            printf("array1[%u] = %d ", j, array1[j] += 3);
        }
    }
}
```

- A) 0 0 0 0 0
- B) 5 5 5 5 5
- C) 5 10 15 20 25
- D) 15 15 15 15 15
- E) 25 25 25 25 25

T6. Aşağıda verilen kodun çıktısı ne olabilir? (2023 verisinin, bilgisayarın “0062FE1C” nolu bellek adresine atandığını varsayalım.)

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

```
int main(void){
    int a = 2023;
    int *aPtr = &a;
    printf("\n%p\n%p", &aPtr, *&aPtr);
}
```

- A) 2023
2023
- B) 2023
0062FE1C
- C) 0062FE1C
2023
- D) 0062FE1C
0062FE1C
- E) Hiçbiri

T7. Aşağıda verilen kodun çıktısı ne olabilir? (6 verisinin, bilgisayarın “0072FFFC” nolu bellek adresine atandığını varsayalım.)

```
#include <stdio.h>

void Reference(int *nPtr);

int main(void){
    int number = 6;
    Reference(&number);
    printf("\n%d", number);
}

void Reference(int *nPtr){
    *nPtr = *nPtr ;
}
```

- A) 6
- B) 6 0072FFFC
- C) 0072FFFC 6
- D) 0072FFFC
- E) 36

T8. Aşağıdaki programın çıktısı ne olabilir?

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

int main(void){
    const int string1 = 2023;
    const char *string2 = "ISTE_Blg_Muh";
    const char *string = "2023ISTE_ Muh";
    char *remainderPtr;
    unsigned long int x = strtoul(string, &remainderPtr, 0);
    printf("%i\n%s\n", x, remainderPtr);
}
```

- A) 2023
ISTE_ Muh
- B) 2023
ISTE_Blg_Muh
- C) ISTE_Muh
- D) 2023
ISTE_Muh
- E) 2023
ISTE_Blg_Muh

T9. Aşağıdaki program çalıştırıldığında ve ekrana “bilgisayar” kelimesi girildiğinde programın çıktısı ne olur?

```
#include <stdio.h>

int main(void){
    char z[9];
    printf("%s", "Bir metin giriniz ");
    scanf("%8[bilgsyr]", z);
    printf("The input was \"%s\"\n", z);
}
```

- A) bilgisayar
- B) bilg
- C) b
- D) iiaa
- E) bilgis

T10. Aşağıdaki program çalıştığında program çıktısı ekran görüntüsü ne olur?

```
#include <stdio.h>

struct deste {
    char *kart;
    char *tip;
};

int main(void){
    struct deste kart_tipi;
    kart_tipi.kart = "Vale";
    kart_tipi.tip = "Sinek";
    struct deste *destePtr = &kart_tipi;
    printf("%s%s\n%s%s\n", kart_tipi.kart, " - ", kart_tipi.tip,
    destePtr->kart, " - ", destePtr->tip);
}
```

A) Vale - Sinek

Vale – Sinek

B) Sinek – Vale

C) Vale - Sinek

Vale - Sinek

Vale - Sinek

D) Vale – Sinek

E) Vale ve Sinek kelimelerinin Bellek adresleri görünür ve her bilgisayara göre değişir.

S-2) Aşağıda verilen özellikleri sağlayan bir C kodu yazınız.

- 20 elemanlı bir tamsayı dizisi tanımlanacak.
- 20 tam sayının klavyeden rastgele girilmesi istenecek
- Klavyeden girilen elemanlar büyükten küçüğe doğru Bubble sort algoritması ile sıralanacak ve yazdırılacak.

S-3) Klavyeden girilen 2 pozitif tam sayının Binary karşılığını 16 bit formatında (dörtlü gruplar halinde) yazdıran ve bu sayıların Binary eşdeğerlerini Özel Veya (Exclusive OR) işlemine tabi tutup sonucu hem binary hem de tamsayı karşılığını yazan bir program yapınız.

S-4) Kullanıcıdan **email adresi** ve **öğrenci numarası** (ID) bilgileri istenmektedir. Bu bilgilerin doğruluğu, programda tanımlı olan doğru bilgilerle karşılaştırılarak kontrol edilmektedir. Doğru bilgiler en başta tanımlanmalı ve herkesin kendi **eposta adresi** ve **öğrenci numarası** olmalıdır.

Örnek:

```
char correct_email[100] = "batuhan@karadag";  
char correct_id[100] = "1234567890";
```

Programın Özellikleri:

- **input()** adında bir fonksiyon, fonksiyon pointer'ı (işaretçisi) kullanarak hangi input fonksiyonunun çağrılacağını belirler.
- **inputEmail()** fonksiyonu, kullanıcıdan email adresini alır ve 5 karakterden kısa olması durumunda tekrar sorar. 5 karakterden fazla olacak kadar girdi istenir.
- **inputID()** fonksiyonu, kullanıcıdan öğrenci numarasını (ID) alır ve 10 haneli değilse tekrar sorar. 10 haneden fazla olacak kadar girdi istenir.
- **stringLength()** fonksiyonu, girilen string'in uzunluğunu hesaplar.
- **validate()** fonksiyonu, girilen email ve öğrenci numarasını doğru tanımlı bilgilerle karşılaştırır.
 - Eğer her ikisi de doğru ise **“Giriş başarılı! Hoş geldin, [email]”** mesajı verir.
 - Yanlışsa **“Giriş başarısız! Lütfen bilgilerinizi kontrol ediniz.”** mesajı gösterilir.

Önemli Notlar:

- Aşağıda gerekli kütüphaneler ve global değişkenler verilmiştir. Fonksiyon prototipleri yazılmalıdır. (8p)
- **Fonksiyon pointer'ı** mutlaka kullanılmalı ve input() fonksiyonu aracılığıyla input alınmalıdır. (13p)
- Programın sonunda doğrulama yapılmalı ve sonucu ekrana yazdırmalıdır. (6p)
- Kodunuzda yorum satırları (*// açıklamalar*) kullanarak fonksiyonların görevlerini belirtiniz. (3p)

```
#include <stdio.h>  
#include <string.h>  
  
// Global değişkenler  
char correct_email[100] = ".....@.....";  
char correct_id[100] = ".....";  
  
// Kullanıcıdan alınacak inputlar  
char input_email[100];  
char input_id[100];
```