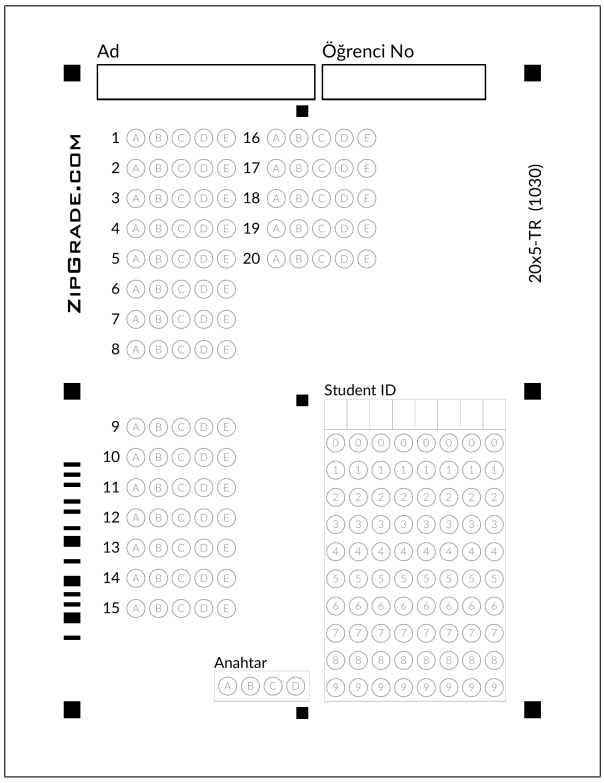
BLM5230 İleri Programlama Teknikleri Final Sınavı
2023 - 2024 - 1, 15 Ocak 2024, Anahtar A



Notes:

- Cevaplarınızı cevap anahtarına işaretleyiniz. Cevap anahtarı bölgesi içine cevap işaretlemeleri haricinde yazım/çizim yapmayınız, optik okuyucu programı şaşırtmaktadır.
- Karalama kağıdı kullanılabilir. Karalama kağıdınızı sınav sonunda teslim ediniz. Dışarı çıkarmak yasaktır.
- Soru kağıdına karalama yapabilirsiniz, ancak cevabınızı belli edecek herhangi bir işaretleme yapmayınız. Bu şekildeki işaretlemeler kopya verme girişimi muamelesi görecektir (sorudan alınan puan iptal edilecektir). Aynı şekilde karalama kağıdında bir soru ile cevabını eşleyen bir yazma/işaretleme yapmayınız. Aynı şekilde muamele görecektir.
- İsminizi ve numaranızı cevap anahtarında ilgili alanlara yazınız ve işaretleyiniz. Cevap anahtarında 'Anahtar' alanınını doğru işaretlediğinizden emin olunuz. Sınav optik olarak okunacaktır.
- İngilizce/Türkçe bilmediğiniz kelimeleri sorabilirsiniz. Anlamadığınız soruları sorabilirsiniz.
- Yanlış sorular olabilir. Sınav sorumlusuna şüphelendiğiniz soruları bildiriniz. Sınav sonrasında kontrol edilecek ve hata var ise iptal edilecektir. Yine bazen aynı cevap sehven birden fazla şıkka basılmaktadır. Bu durumda alfabetik olarak önce gelen şıkkı işaretleyiniz ve sınav görevlilerini durumdan haberdar ediniz. Sınav görevlileri okuyucu programı uygun ayarlayacaktır.
- Hesap makinesi kullanılabilir.
- Tüm sorular eşit puanlıdır.

Questions (Each equal in points):

Aşağıdaki `Makefile` alıntısında belirtilen hedefler(target), kaynak dosyaları(source files) ve komutlar(command) ilişkisini tanımlayan açıklamalardan hangisi yanlıştır?

```
all: program

program: main.o utils.o
    gcc -o program main.o utils.o

main.o: main.c utils.h
    gcc -c main.c

utils.o: utils.c utils.h
    gcc -c utils.c
```

A. `all` hedefi, `program` hedefine bağlıdır ve `gcc` komutu kullanılarak tüm uygulama oluşturulur.

B. `program` hedefi, `main.o` ve `utils.o` nesne dosyalarına bağlıdır ve bu dosyalar eklenerek program oluşturulur.

C. `main.o` ve `utils.o` hedefleri, karşılıklı `.c` ve `.h` dosyalarına bağlıdır ve `gcc -c` komutuyla derlenirler.

D.

1. Aşağıdaki C programı kod bloklarından hangisi 'dangling else problemi' (asılı else problemleri) engellemek için doğru bir yaklaşım sunar?

   if (x > 0) if (y > 0) printf("x ve y pozitif."); else printf("y negatif veya 0.");if (x > 0) if (y > 0) printf("x ve y pozitif."); else printf("y negatif veya 0."); if (x > 0) if (y > 0) printf("x ve y pozitif."); else printf("y negatif veya 0.");

A.

3. C dilinde, string sabitine atanan bir char işaretçisi ile yapılan hangi işlem hata sonucuyla karşılaşır?

   String sabitine atanan işaretçi ile dizi elemanlarının adreslerini okumak.String sabitini başka bir işaretçiye yeniden atamak. String sabitine atanan işaretçi ile string'in karakterlerini değiştirmeye çalışmak.

A.
```
struct personalstat {
char ps_name[20], ps_tcno[11];
unsigned int ps_birth_day : 5;
unsigned int ps_birth_month : 4;
unsigned int ps_birth_year : 11;
struct personalstat *next;
};
typedef struct personalstat ELEMENT;
static ELEMENT *head;
```

   Yukarıdaki C yapı (struct) tanımında, 'ps_birth_day', 'ps_birth_month' ve 'ps_birth_year' alanları bit alanları (bit fields olarak) tanımlanmıştır. Bu alanlardan her biri için MAKSİMUM saklanabilecek değer nedir?

   A. ps_birth_day: 2^5, ps_birth_month: 2^4, ps_birth_year: 2^{11}

   B. ps_birth_day: $2^5 - 1$, ps_birth_month: $2^4 - 1$, ps_birth_year: $2^{11} - 1$

   C. ps_birth_day: 31, ps_birth_month: 12, ps_birth_year: 2047

   D.

5. Hangisi doğrudur?

   A. a <<= b ifadesi, a'nın bitlerini sol tarafa doğru b kadar kaydırır ve sonucu a'nın yerine atar.

   B. a += b ifadesi yalnızca tam sayılar için kullanılır ve a'nın değerine b'nin değerini ekler.

   C. a ||= b ifadesi, a'nın değerini veya b'nin değerini a'nın yerine atar.

   D.

6. Aşağıdakilerden hangisi iş parçacıkları (threads) ve işlemler (processes) arasındaki farklar ile ilgili olarak doğru bir ifade değildir?

   Her iş parçacığı kendi adres alanını ve açık dosya tanımlayıcılarını oluşturur, bu da işlemler arasında bellek paylaşımı olmadığını gösterir.İş parçacıkları, aynı işlem altında daha hızlı bağlam değişikliği yapabilirken, işlemler arasındaki bağlam değişikliği daha yavaş olur. İşlemler birbirinden bağımsız çalışır ve işlemler arası senkronizasyon sadece işletim sistemi tarafından gerçekleştirilir; buna karşılık, iş parçacığı senkronizasyonu altındaki işlem tarafından yönetilir.

A. Aşağıdaki hangi işlem geçersizdir? ('int ar[4];' tanımı yapılmıştır ve `p` bir tam sayı pointerıdır.)

A. `*p = 5;` B. `ar++;` C. `p++;` D. `p = ar;`

8. C programlama dilindeki rastgele erişim (Random Access) işlevlerinden fseek() ve ftell() kullanılarak bir dosyanın tam ortasında bulunan byte değerine erişilmek istendiğinde, aşağıdaki seçeneklerden hangisi doğru kullanımdır?

   stat = fseek(fp, -10, SEEK_END);stat = fseek(fp, ftell(fp)/2, SEEK_CUR); pos = ftell(fp); fseek(fp, pos / 2, SEEK_SET); fseek(fp, 10, 0);

A.

10. Aşağıdaki C kodu verilmiştir:

```
#include <stdio.h>

int main() {
    int numbers[] = {10, 20, 30, 40, 50};
```

```
    int *ptr = numbers;
    int result;

    result = sizeof(ptr) < sizeof(*ptr) ? sizeof(numbers) : sizeof(numbers[0]);
    printf("%d", result);
    return 0;
}
```

Adreslerin 8, tamsayıların ise 4 byte hafıza aldığı bir sistemde bu kod parçacığı konsola neyi basar?

A. `sizeof(numbers[0])`

B. `sizeof(numbers)`

C.

11. Tuple yapılarında (*Tuple Structs*) aşağıdaki Rust kod parçasının amacı nedir?

```
struct Meters(i32);
struct Yards(i32);
```

Aynı yapısal özelliklere sahip oldukları halde, bu yapılar birbiriyle eşitlenemez.Bu yapılar, aynı yapısal özelliklere sahip ve birbiriyle eşitlenebilir olan yeni tipler (type) tanımlar. Bu yapılar, özdeş (*identical*) fonksiyonlar için farklı adlar sağlar.

12. Aşağıdakilerden hangisi dizilerin (arrays) sıralama (sorting) işlemlerinde işlevlerin (functions) kullanılmasının avantajlarını açıklar?

A. Sıralama işlemi sırasında dizilerin hafızada daha az yer kaplamasını sağlar.

B. Kod tekrarını azaltır ve kodun bakımını kolaylaştırır.

C. Sıralama algoritmalarının performansını artırır

D.

13. Aşağıdaki C ifadelerinde verilen değişkenler ve işlemleri dikkate alarak, '**p' ifadesinin değerinin ne olacağını belirtiniz.

```
int r = 5;
int *q = &r;
int **p = &q;
r = 10;
```

A. 5 B. 10 C. Bir adres değeri D. Hiçbiri

14. Unit yapıları (*Unit Structs*) veya sıfır boyutlu tipler (*Zero-Sized Types*) ile ilgili olarak, aşağıdaki Rust kod parçasındaki yapı ne amaçla kullanılır?

```
struct Marker;
```

Bir fonksiyonun dönüş tipi olarak kullanılır, hiçbir değer dönmediğini gösterir.Bir veri yapısının (data structure) içerisinde yer ayırtmak için kullanılır.

15. Rust dilinde veri sahipliğinin geçici olarak ödünç alınması (borrowing) durumunda, aşağıdakilerden hangisi doğrudur?

Özgün (original) değişken sahipliğini ödünç alma süresince ve sonrasında korur.Ödünç alınan değişken üzerinde sınırsız değişiklik yapılabilir. Sahipliliği ödünç alınan değişken artık kullanılamaz (impossible to use).

16. Aşağıdaki ifadelerden hangisi C programlama dilinde sabit tanımı (constant definition / macro definition) ile ilgili olarak doğrudur?

Sabit ismi tanımlarken, tam sayı değerlere açıklayıcı isimler vermek ve programı değiştirmeyi kolaylaştırmak için #define önişlemci direktifi kullanılır. Örneğin, '#define NOTHING 0'.Sabitlerin (constants) değişken olarak kullanılması yaygın bir uygulamadır, örneğin 'NOTHING = j + 5;'. #include <stdio.h> direktifi, işletim sistemi veya C kütüphanesi tarafından tanımlanmamış özel bir dosyayı include etmek için kullanılır.

17.