S-1) Aşağıdaki test sorularından doğru olanı işaretleyiniz. (Her test 3 puan. Toplam 30 puan)

T1) Malloc() fonksiyonu ile bellekte kullanılan alanı serbest bırakmak için aşağıdakilerden hangisi kullanılmalıdır?

A) Remove()

B) Free()

C) Delete()

D) Calloc()

E) Realloc()

Cevap B. Free() fonskiyonu ile tahsis edilen alan serbest bırakılır.

T2) n1 ve n2 yapılarına ait bilgiler aşağıdaki gibidir. **char** veri tipi , hafızada 1 byte yer kapladığına göre n1 ve n2 yapıları kaç byte yer kaplar?

union n1{

char a[5];

char b[10];

};

struct n2{

char a[5];

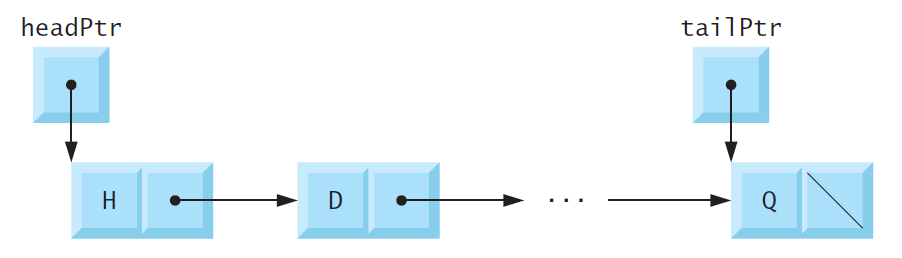
char b[10];

};

1. n1 : 15 , n2: 15
2. n1 : 15 , n2: 10
3. n1 : 10 , n2: 15
4. n1 : 0 , n2: 0
5. N1: 10 , n2:10

Cevap: C union yapısı içerisindeki en büyük olan kadardır. Sturuct yapısı içersisindekilerin toplamı kadardır.

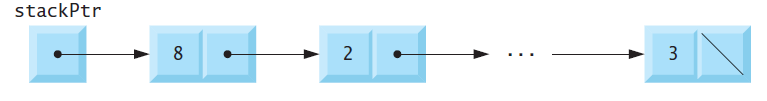
T3) Aşağıdaki şekilde verilen bir kuyruk yapısına sırasıyla B L G M U H karakterleri giriliyor. Bu kuyruk yapısına peşpeşe 2 defa “dequeue” fonksiyonu uygulanırsa kuyruk yapısının son durumu ne olur?



1. L G M U b) G M U H c) B L G M d) B L U H e) B L G M U H

Cevap : b

T4) Aşağıdaki diyagram, birkaç düğüme sahip bir yığın yapısını göstermektedir; stackPtr, yığının en üst öğesini gösterir.



Bir yığın yapısına 6 defa itme (push) fonksiyonu ile sırasıyla 1 2 3 4 5 ve 6 verileri ekleniyor. Daha sonra bu yığın yapısına 2 defa çekme (pop) fonksiyonu uygulanırsa belleğin son durumu ne olur.

1. 3 4 5 6 b) 1 2 5 6 c) 2 3 4 5 d) 1 2 3 4 e) 1 2 3 4 5 6

Cevap: d

T5) Dinamik bellek ayırma fonksiyonları kullanılarak oluşturulan bir “bağlantılı liste (linked list)” yapısı için aşağıda verilen açıklamalardan hangisi yanlıştır.

1. Bağlantılı bir liste, düğüm adı verilen ve işaretçi bağlantılarıyla birbirine bağlanan, kendine referanslı yapıların doğrusal bir koleksiyonudur. Listenin ilk düğümüne atanan bir işaretçi aracılığıyla erişilir. Sonraki düğümlere, her düğümde bulunan bağlantı işaretçisi aracılığıyla erişilir.
2. Bir düğüm, diğer struct yapıları da dahil olmak üzere herhangi bir türde veri içerebilir. Bağlantılı listeler doğrusal veri yapılarıdır.
3. Bağlantılı listeler dinamiktir, bu nedenle bir listenin uzunluğu yürütme sırasında gerektiği gibi artabilir veya azalabilir.
4. Bağlantılı liste düğümleri bellekte bitişik olarak depolanır. Veriler sırayla eklenir silinirken de listenin başından veya sonundan sırayla silinebilir.
5. Liste yapısına “function insert” ile bir karakter eklenebilir, yada “function delete” ile bir karakter silinebilir.

Cevap: d

T6) Rastgele erişimli dosyalar ile ilgili verilen tanımlardan hangisi yanlıştır.

a) Rasgele erişimli bir dosyaya yazdığınız ve buradan okuduğunuz bireysel kayıtların uzunluğu sabittir ve diğer kayıtlar aranmadan doğrudan erişilebilir.

b) Bir rasgele erişim dosyasındaki bir kaydın dosyanın başlangıcına göre tam konumu, kayıt anahtarının bir fonksiyonu olarak hesaplanabilir.

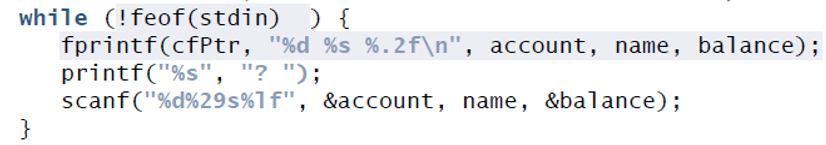
c) “fwrite” fonksiyonu, bellekte belirli bir konumdan başlayarak belirli sayıda baytı bir dosyaya aktarır. Veriler, dosya konumu işaretçisi tarafından belirtilen dosya konumundan başlayarak yazılır.

d) “fwrite” ve “fread” fonksiyonları dosyalara veri dizilerini yazabilir ve dosyalardan okuyabilir. Hem fread hem de fwrite'ın üçüncü argümanı, dizideki bir dosyadan okunması veya dosyaya yazılması gereken öğelerin sayısıdır.

e) Bir rasgele erişim dosyasındaki her kayıt normalde aynı uzunluğa sahip olduğundan, sırasal erişimli dosyalara göre bellekte daha az yer kaplar.

Cevap e

T7) C’de yazılmış sırasal erişimli bir dosya oluşturma programında aşağıda verilen program parçası ile ilgili açıklamalardan hangisi yanlıştır.



a) Dosya sonu komutuna erişilinceye kadar veri okumaya devam eder.

b) account, name, balance değişkenlerine sırası ile integer, string, double formatlarında veri girer.

c) Daha önce tanımlanmış “stdin” değerine ulaşıldığında while döngüsünden çıkar.

d) account, name, balance değişkenlerine girilen değerleri ekrana yazar.

e) Dosya sonu için ctrl+z kullanılır.

Cevap c

T8) Aşağıdaki programla ilgili seçeneklerden hangisi yanlıştır.

#include <stdio.h>

struct card {

char \*face;

char \*suit;

};

int main(void)

{

struct card aCard;

aCard.face = "BLG";

aCard.suit = "MUH";

struct card \*cardPtr = &aCard;

printf("%s %s", aCard.face, aCard.suit);

}

1. Ekrana “BLG MUH” yazdırır.
2. cardPtr işaretçisine “aCard” struct yapısının adresi atanır.
3. aCard yapısının üyelerine sırasıyla "face" ve "suit" değerleri atanır.
4. Bu struct yapısı char türünde veri içerir.
5. printf fonksiyonunda cardPtr->face, cardPtr->suit yazılırsa yine aynı sonuç elde edilir.

Cevap: C

T9) Aşağıdaki program ile ilgili hangi açıklama doğrudur.

#include <stdio.h>

int main(void)

{

int a = 74;

int \*aPtr = &a;

printf("%p\n",aPtr);

}

1. aPtr işaretçisinin adresini yazdırır.
2. a değişkeninin değerini yazdırır.
3. a değişkeninin adresini yazdırır.
4. aPtr işaretçisinin değerini yazdırır.
5. Hiçbiri

Cevap :c

T10) Sırasal Erişimli ve rastgele erişimli dosya işlemleri ile ilgili hangi seçenek yanlış olur.

1. Rastgele erişimli dosyalarda veri girişi için ayrılan bellek hücreleri sabit uzunlukludur.
2. Bir dosyadan sıralı olarak veri almak için, bir program dosyanın başından okumaya başlar ve istenen veri bulunana kadar tüm verileri ardışık olarak okur.
3. Bir dosyayı herhangi bir kipte açarken bir hata oluşursa, fopen NULL değerini döndürür.
4. Sabit uzunluklu kayıtlar, dosyadaki diğer verileri yok etmeden verilerin rastgele erişimli bir dosyaya eklenmesini sağlar.
5. Sırasal erişimli bir dosyada kısa bir veri daha uzun bir veri ile veri kaybı olmadan değiştirilebilir.

Cevap: e

S-2) “srand” ve “time” fonksiyonlarını kullanarak 100 elemanlı, 1 ile 50 arası rastgele sayılardan oluşan bir dizi elde ediniz. Elde ettiğiniz dizide rastgele üretilen her sayıdan kaç tane üretildiğini hesaplayan ve bunları ekrana yazdıran bir program yazınız.

CEVAP 2

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int main(){

int dizi[100];

int frekans[51]={0};

int i;

srand(time(NULL));

for(i=0; i<100; i++){

dizi[i]=1+rand() %51;

frekans[dizi[i]]++;

printf("%d \t", dizi[i]);

}

puts("");

printf("Dizinin elemanlari ve frekanslari\n");

for(i=0; i<51; i++){

if (frekans[i]>0) {

printf("%d Sayisi %d defa uretildi\n", i, frekans[i]);

}

}

}

S-3) Kendi isminiz ile bir \*.txt dosyası açan (örneğin: Batuhan.txt), bu dosyaya hesap no (int), isim (char) ve hesaptaki miktarı (double) girmemizi sağlayan bir C programı yazınız. Bu dosyaya istenildiği kadar kayıt girilebilecek, kayırt girme işlemini sonlandırmak için EOF kullanılacak. Kayıt girme işlemi bitince dosya kapatılacak.

CEVAP 3

#include <stdio.h>

int main(void)

{

FILE \*cfPtr;

if ((cfPtr = fopen("Batuhan.txt", "w")) == NULL) {

puts("Dosya acilamiyor");

}

else {

puts("Hesap no, isim, ve hesaptaki miktari giriniz.");

puts("Bitirmek icin EOF giriniz.");

unsigned int account; // hesap no

char name[30]; // isim

double balance; // hesaptaki para

scanf("%d%29s%lf", &account, name, &balance);

while (!feof(stdin) ) {

fprintf(cfPtr, "%d %s %.2f\n", account, name, balance);

printf("%s\n", "");

scanf("%d%29s%lf", &account, name, &balance);

}

fclose(cfPtr);

}

}