

PRM kolokwium 2

Czas pisania kolokwium **50 minut**

Do uzyskania w sumie **30 punktów**

Proszę odpowiadać na zadane pytania i rozwiązywać zadania według treści przekazanej na kartce. Zrozumienie treści zadania jest elementem rozwiązania. Dopuszczalne jest korzystanie z notatek i książek, żadne urządzenia elektroniczne nie są dozwolone. Punktacja jest określona przy numerze zadania.

Proszę uzupełnić dwa pierwsze pola tabelki obok. Zadań proszę nie przepisywać. Wszystkie oddawane kartki powinny być podpisane własnym imieniem i nazwiskiem. Proszę wyraźnie zaznaczyć odpowiedzi. Nie uzupełnienie tabelki powoduje ocenę 0 z kolokwium. Kartki nie podpisane nie będą sprawdzane.

A
imię, nazwisko
grupa studencka
prowadzący laboratorium
Zad.1
Zad.2
Zad.3
Suma

Zad 1(18)

Załóżmy, że mamy listę liniową, której kostka składowa jest postaci

```
struct wezel{
    int info;
    struct wezel *nast;
};
```

Pole `nast` w zawiera wskaźnik następnego elementu listy. Jeśli taki element nie istnieje, wówczas ma ono wartość `NULL`.

Proszę zaimplementować funkcje `push` i `pop` realizujące kolejkę FIFO używając kostek typu `struct wezel`. Wskazówka: proszę zastanowić się nad właściwą definicją punktu wejścia (punktów wejścia) do kolejki.

PRM kolokwium 2

Czas pisania kolokwium **50 minut**

Do uzyskania w sumie **30 punktów**

Proszę odpowiadać na zadane pytania i rozwiązywać zadania według treści przekazanej na kartce. Zrozumienie treści zadania jest elementem rozwiązania. Dopuszczalne jest korzystanie z notatek i książek, żadne urządzenia elektroniczne nie są dozwolone. Punktacja jest określona przy numerze zadania.

Proszę uzupełnić dwa pierwsze pola tabelki obok. Zadań proszę nie przepisywać. Wszystkie oddawane kartki powinny być podpisane własnym imieniem i nazwiskiem. Proszę wyraźnie zaznaczyć odpowiedzi. Nie uzupełnienie tabelki powoduje ocenę 0 z kolokwium. Kartki nie podpisane nie będą sprawdzane.

B	
imię, nazwisko	
grupa studencka	
prowadzący laboratorium	
Zad.1	
Zad.2	
Zad.3	
Suma	

Zad 1(18)

Załóżmy, że mamy listę liniową, której kostka składowa jest postaci

```
struct wezel{
    int info;
    int prior;
    struct wezel *nast;
};
```

Pole `nast` w zawiera wskaźnik następnego elementu listy. Jeśli taki element nie istnieje, wówczas ma ono wartość `NULL`.

Proszę zaimplementować funkcje `push` i `pop` realizujące kolejkę priorytetową (tzn. posortowaną według wartości priorytetu) używając kostek typu `struct wezel`.

Wskazówka: proszę zastanowić się nad właściwą definicją punktu wejścia do kolejki.

PRM kolokwium 2

Czas pisania kolokwium **50 minut**

Do uzyskania w sumie **30 punktów**

Proszę odpowiadać na zadane pytania i rozwiązywać zadania według treści przekazanej na kartce. Zrozumienie treści zadania jest elementem rozwiązania. Dopuszczalne jest korzystanie z notatek i książek, żadne urządzenia elektroniczne nie są dozwolone. Punktacja jest określona przy numerze zadania.

Proszę uzupełnić dwa pierwsze pola tabelki obok. Zadań proszę nie przepisywać. Wszystkie oddawane kartki powinny być podpisane własnym imieniem i nazwiskiem. Proszę wyraźnie zaznaczyć odpowiedzi. Nie uzupełnienie tabelki powoduje ocenę 0 z kolokwium. Kartki nie podpisane nie będą sprawdzane.

C
imię, nazwisko
grupa studencka
prowadzący laboratorium
Zad.1
Zad.2
Zad.3
Suma

Zad 1(16)

Rozważmy wyrażenia regularne, które składają się z wyłącznie z małych liter oraz znaków '*' (zastępujących dowolną liczbę wystąpień dowolnej małej litery) i '?' (zastępujących jedną dowolną małą literę). Napisać funkcję

```
int pasuje(char* wzorzec, char* napis);
```

która zwraca zero, gdy napis nie pasuje do wzorca opisanego wyrażeniem regularnym, zaś jeden – gdy takie dopasowanie ma miejsce. Przykłady użycia funkcji

```
pasuje("a*ama*", "alabama") -> zwraca wartość 1
```

```
pasuje("?a*ma*", "mama") -> zwraca wartość 1
```

```
pasuje("?a*a*", "ala") -> zwraca wartość 0
```

Proszę pamiętać, że łańcuch znakowy jest tablicą wartości typu char, a koniec łańcucha jest oznaczany wartością 0 typu char.

PRM kolokwium 2

Czas pisania kolokwium **50 minut**

Do uzyskania w sumie **30 punktów**

Proszę odpowiadać na zadane pytania i rozwiązywać zadania według treści przekazanej na kartce. Zrozumienie treści zadania jest elementem rozwiązania. Dopuszczalne jest korzystanie z notatek i książek, żadne urządzenia elektroniczne nie są dozwolone. Punktacja jest określona przy numerze zadania.

Proszę uzupełnić dwa pierwsze pola tabelki obok. Zadań proszę nie przepisywać. Wszystkie oddawane kartki powinny być podpisane własnym imieniem i nazwiskiem. Proszę wyraźnie zaznaczyć odpowiedzi. Nie uzupełnienie tabelki powoduje ocenę 0 z kolokwium. Kartki nie podpisane nie będą sprawdzane.

D	
imię, nazwisko	
grupa studencka	
prowadzący laboratorium	
Zad.1	
Zad.2	
Zad.3	
suma	

Zad 1(16)

Proszę napisać funkcję o nagłówku

```
char* maxWspPodciag(char* a, char* b);
```

która wyszukuje w ciągach znaków *a*, *b* podanych jako argumenty, wspólny ciąg o największej długości. Gdy ciąg ten zostanie znaleziony, funkcja wykonuje jego kopię i zwraca ją jako wynik działania; w przeciwnym przypadku, zwracana jest wartość NULL. Wskazówka: można korzystać z biblioteki `string`.

Proszę pamiętać, że łańcuch znakowy jest tablicą wartości typu `char`, a koniec łańcucha jest oznaczany wartością 0 typu `char`.

Zad 3(12)

Jaki będzie stan standardowego wyjścia w wyniku uruchomienia poniższego programu?
Odpowiedź proszę krótko uzasadnić.

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

int x[8]={3,6,-2,5,4,8,7,-5};

int max(int a, int b)
{
    if (a>b)
        return a;
    else
        return b;
}

int fun(int *x, int n)
{
    if (n>=2)
        return max(fun(x,n/2), fun(x+n/2, n-n/2) ) ;
    else if (n==1)
        return *x;
    else
        return -INT_MAX;
}

int main()
{
    int x=7;
    int tab[10]={2,-3, 5, -10, 6, 7, 0, 5, 20, -80};

    printf("%d\n", fun(tab,8));

    return 0;
}
```

Zad 3(12)

Jaki będzie stan standardowego wyjścia w wyniku uruchomienia poniższego programu?
Odpowiedź proszę krótko uzasadnić.

```
int x[8]={3,6,-2,5,4,8,7,-5};

int min(int a, int b)
{
    if (a<b)
        return a;
    else
        return b;
}

int fun(int *x, int n)
{
    if (n>=2)
        return min(fun(x,n/2), fun(x+n/2, n-n/2) ) ;
    else if (n==1)
        return *x;
    else
        return INT_MAX;
}

int main()
{
    int x=7;
    int tab[10]={2,-3, 5, -10, 6, 7, 0, 5, 20, -80};

    printf("%d\n", fun(tab,8));

    return 0;
}
```

Zad 3(14)

Jaki będzie stan standardowego wyjścia w wyniku uruchomienia poniższego programu?
Odpowiedź proszę krótko uzasadnić, szkicując sposób powstawania struktury dowiązań wskaźnikowych.

```
struct s{
    int x;
    struct s* y;
};

struct s* fun1(struct s* k, int x)
{
    struct s* w;
    w = (struct s*) malloc(sizeof(struct s));
    w->x=x;
    w->y=NULL;
    if (k!=NULL)
        k->y=w;
    return w;
}
struct s* fun2(struct s* p, int *x)
{
    struct s* u;
    if (p!=NULL)
    {
        *x=p->x;
        u=p; p=p->y;
        free(u);
    }
    return p;
}
int main()
{
    struct s *p=NULL, *k=NULL;
    int x;

    p=k=fun1(k,3);
    k=fun1(k,5);
    k=fun1(k,7);
    p=fun2(p, &x);
    printf("%d\n", x);
    k=fun1(k,9);
    p=fun2(p, &x);
    printf("%d\n", x);
    return 0;
}
```

Zad 3(14)

Jaki będzie stan standardowego wyjścia w wyniku uruchomienia poniższego programu?
Odpowiedź proszę krótko uzasadnić, szkicując sposób powstawania struktury dowiązań wskaźnikowych.

```
struct s{
    int x;
    struct s* y;
};

struct s* fun1(struct s* p, int x)
{
    if ((p!=NULL)&&(p->x>x))
        p->y=fun1(p->y,x);
    else
    {
        struct s* w;
        w = (struct s*) malloc(sizeof(struct s));
        w->x=x;
        w->y=p;
        return w;
    }
}

struct s* fun2(struct s* p, int *x)
{
    struct s* u;
    if (p!=NULL)
    {
        *x=p->x;
        u=p; p=p->y;
        free(u);
    }
    return p;
}

int main()
{
    struct s *p=NULL;
    int x;

    p=fun1(p,3);
    p=fun1(p,5);
    p=fun1(p,7);
    p=fun2(p, &x);
    printf("%d\n", x);
    p=fun1(p,9);
    p=fun2(p, &x);
    printf("%d\n", x);
    return 0;
}
```