

Egzamin (przykład)

imię i nazwisko _____

liczba punktów _____ / 20

ocena _____

punkcja

5.0 [19, 20]

4.5 [17, 19)

4.0 [15, 17)

3.5 [12, 15)

3.0 [11, 12)

2.0 [0, 11)

dokument klasy B1, archiwizować do 2022-01-01

Dla zadań 1, 2 i 3 zdefiniowano strukturę:

```
1 struct element
2 {
3     int wartosc;
4     element * pNext;
5 };
```

Dla zadań 4 i 5 zdefiniowano strukturę:

```
1 struct wezel
2 {
3     int wartosc;
4     wezel * pLewy, * pPrawy;
5 };
```

-
1. (2) Proszę zaimplementować rekurencyjną funkcję **int** zsumuj (**element** * pHead); która sumuje elementy listy. Jeżeli lista jest pusta, funkcja zwraca zero.

2. (2) Proszę zaimplementować **void** odwroc (**element** * & pH); która odwraca kolejność elementów w liście jednokierunkowej.

- (2) Proszę zaimplementować funkcję, która zwraca liczbę elementów listy cyklicznej (w której następnikiem ostatniego elementu jest element pierwszy).
- (2) Proszę zaimplementować funkcję, która wyznaczy wysokość drzewa, czyli maksymalną liczbę węzłów od korzenia do liścia. Puste drzewo ma wysokość 0.
- (2) Proszę napisać funkcję, która liczy, ile węzłów w drzewie ma dwa potomki.

6. (2) Proszę napisać funkcję, która odczytuje liczby całkowite z pliku o podanej nazwie. Funkcja zwraca wektor odczytanych liczb. Jeżeli pliku nie da się otworzyć lub jest pusty, funkcja zwraca pusty wektor.
7. (2) Proszę napisać funkcję, która przyjmuje tablicę wartości typu **int** i zwraca adres zaalokowanej tablicy dwa razy dłuższej zawierającej wartość oryginalną i jej kwadrat. Przykład: dla tablicy { 2, 1, -3, -4} funkcja zwróci adres tablicy zawierającej wartości {2, 4, 1, 1, -3, 9, -4, 16}.

8. (2) Proszę napisać funkcję `kumulacja`, która przyjmuje tablicę i dla elementu a_i tablicy wyznacza wartość $\sum_{l=0}^i a_l$ i wpisuje tę wartość do przekazanej tablicy (funkcja modyfikuje zawartość tablicy).

9. (2) Proszę napisać makrodefinicję `swap(t,x,y)`, zamieniającą miejscami wartości przechowywane w zmiennych `x` i `y` typu `t`.

10. (2) Proszę napisać funkcję `odwroc`, która odwraca kolejność znaków w łańcuchu znakowym typu `char *`.

Rozwiązania

zadanie 1

```
1 int zsumuj (element * pHead)
2 {
3     if (not pHead)
4         return 0;
5     else
6         return pHead->wartosc + zsumuj (pHead->pNext);
7 }
```

zadanie 2

```
1 void odwroc (element * & pH)
2 {
3     element * p = nullptr;
4     while (pH)
5     {
6         auto q = pH->pNext;
7         pH->pNext = p;
8         p = pH;
9         pH = q;
10    }
11    pH = p;
12 }
```

zadanie 3

```
1 int dlugosc (element * pHead)
2 {
3     if (not pHead)
4         return 0;
5     auto p = pHead;
6     int licznik = 1;
7     while (p->pNext != pHead)
8     {
9         p = p->pNext;
10        licznik++;
11    }
12    return licznik;
13 }
```

zadanie 4

```
1 int wysokosc (wezel * pR)
2 {
3     if (not pR)
4         return 0;
5     else
6         return std::max (wysokosc(pR->pLewy), wysokosc(pR->pPrawy)) + 1;
7 }
```

zadanie 5

```
1 int dwaPotomki (wezel * pR)
2 {
3     if (pR)
4         return dwaPotomki (pR->pLewy) + dwaPotomki (pR->pPrawy) + (pR->pLewy
5         and pR->pPrawy);
6     else
7         return 0;
8 }
```

```

6         return 0;
7     }

```

zadanie 6

```

1 std::vector<int> wczytaj (const std::string & nazwa_pliku)
2 {
3     std::vector<int> liczby;
4     std::ifstream plik (nazwa_pliku);
5     if (plik)
6     {
7         int liczba;
8         while (plik >> liczba)
9             liczby.push_back(liczba);
10
11         plik.close();
12     }
13     return liczby;
14 }

```

zadanie 7

```

1 int * powiel (int * t, const int N)
2 {
3     int * p = new int[N * 2];
4     for (int i = 0; i < N; i++)
5     {
6         p[2 * i] = t[i];
7         p[2 * i + 1] = t[i] * t[i];
8     }
9     return p;
10 }

```

zadanie 8

```

1 void kumuluj (double * t, const int N)
2 {
3     for (int i = 1; i < N; i++)
4         t[i] += t[i - 1];
5 }

```

zadanie 9

```

1 #define swap(t,x,y) do { t p; p = x; x = y; y = p; } while (0)

```

zadanie 10

```

1 void odwroc (char * s)
2 {
3     int len = strlen (s);
4     char * t = s + len - 1;
5     char c;
6     while (s < t)
7     {
8         c = *s;
9         *s = *t;
10        *t = c;
11        s++;
12        t--;
13    }
14 }

```