Algorytmy i programowanie

Marcin Żurowski

29 marzec 2021

Plan zajęć

Procedura - definicja

```
void f1(){
 cout << "hello" << endl;
}
int main(){
 ...
 f1();
 ...
}
```

Procedura - deklaracja

```
void f1();
int main(){
 ...
 f1();
 ...
}
void f1(){
 cout << "hello" << endl;
}
```

Funkcja

```
int f1();
int main(){
 ...
 int i = f1();
 ...
}
int f1(){
 cout << "hello" << endl;
 return 0;
}
```

Przekazywanie informacji do funkcji - zmienna globalna

```
int a;
int f1();
int main(){
 ...
 a = 4;
 int i = f1();
 ...
}
int f1(){
 cout << a << endl;
 return 0;
}
```

Przekazywanie informacji do funkcji - przekazywanie argumentu przez wartość

```
int f1(int a);
int main(){
 ...
 int i = f1(4);
 ...
}
int f1(int a){
 cout << a << endl;
 return 0;
}
```

Przekazywanie informacji do funkcji - parametr formalny i aktualny

```
int f1(int a);
int main(){
 ...
 int b = 4;
 int i = f1(b);
 cout << b << endl;
 ...
}
int f1(int a){
 cout << ++a << endl;
 return 0;
}
```

Przekazywanie informacji do funkcji - przekazywanie argumentu przez wskaźnik

```
int f1(int *a);
int main(){
 ...
 int b = 4;
 int i = f1(&b);
 cout << b << endl;
 ...
}
int f1(int *a){
 cout << ++(*a) << endl;
 return 0;
}
```

Przekazywanie informacji do funkcji - przekazywanie tablic

```
int f1(int a[], int n);
int main(){
 ...
 int n = 2;
 int t[2] = {0, 0};
 int i = f1(t, n);
 cout << t[1] << endl;
 ...
}
int f1(int a[], int n){
 a[1] = 4;
 return 0;
}
```

1

Zapisać w postaci procedury poniższy algorytm:

```
integer a, b, k
read(a,b)
if a > b
 k = b
else
 k = a
while (a mod k) ≠ 0 or (b mod k) ≠ 0
 k = k - 1
write(k).
```

2

Zapisać w postaci procedury algorytm wyznaczający liczbę największych elementów tablicy dwuwymiarowej.

3

Podać definicję procedury wyznaczającej najmniejszą wartość tablicy `A`, przy czym chcemy, aby procedura sprawdzała dowolny podciąg od `d` do `g` kolejnych elementów tablicy, gdzie $1 \leq$ `d` $\leq$ `g` $\leq$ `n`.

4

W tablicy `A` zawierającej liczby znajdują się dwa uporządkowane (niemalejąco) ciągi na miejscach od indeksu `p` do indeksu `q` i od indeksu `q + 1` do indeksu `r`, gdzie $\texttt{p} \leq \texttt{q} < \texttt{r}$. Zapisać definicję procedury, która scali te dwa ciągi w jeden niemalejący ciąg i umieści go w tablicy `A` na miejscach od `p` do `r`.

5

W tablicy `A` zawierającej liczby znajdują się dwa uporządkowane (niemalejąco) ciągi na miejscach odpowiednio od indeksu `p` do indeksu `q` i od indeksu `q + 1` do indeksu `r`, gdzie $\texttt{p} \leq \texttt{q} < \texttt{r}$. Zapisać definicję procedury, która scali te dwa ciągi w jeden niemalejący ciąg i umieści go w tablicy `A` na miejscach od `A[p]` do `A[r]` bez sprawdzania za każdym razem, czy wszystkie elementy któregoś z ciągów zostały już wzięte pod uwagę.

6

Zadana jest tablica `A[p..r]` zawierająca liczby. Napisać definicję procedury, która dzieli tę tablicę (poprzez przestawienie jej elementów) na dwie tablice `A[p..q - 1]` i `A[q + 1..r]` w ten sposób, że każdy element z pierwszej podtablicy jest nie większy niż element `A[q]`, który z kolei jest mniejszy od każdego elementu z drugiej podtablicy. Obliczenie indeksu `q` ma stanowić część tej procedury podziału.

7

Zapisać definicję funkcji wyznaczającej najmniejszy element tablicy jednowymiarowej.

8

Zapisać definicję procedury `NWD` w postaci procedury funkcyjnej.

9

Dane są dwa wektory `A[1..n]` `B[1..n]` zawierające liczby.
Napisać definicję funkcji logicznej przyjmującej wartość `true` wtedy i tylko wtedy, gdy oba wektory są równe.