

Consultații Pregătire

Concurs Mate-Info UBB & Examen de Admitere

2023

Algoritmi care lucrează pe numere

Asist. Dr. Mihai Andrei
11.03.2023, 13:00-14:15

- 1) Implementați un algoritm care verifică dacă un număr este prim.

Răspuns:

Varianta C++:

```
bool este_prim(int n){
    if (n < 2) return false;
    if (n == 2) return true;
    if (n % 2 == 0) return false;
    for (int i = 3; i * i <= n; i = i + 2){
        if (n % i == 0) return false;
    }
    return true;
}
```

Varianta Pascal:

```
function este_prim(n:integer):boolean;
var
i: integer;
rez: boolean;
begin
    rez := true;
    if (n < 2) then
        rez := false;
    if (n = 2) then
        rez := true
    else
        begin
            if (n mod 2 = 0) then
                rez := false;
        end;
        i := 3;
        while i*i <= n do
            begin
                if (n mod i = 0) then
                    rez := false;
                i := i + 2;
            end;
        este_prim := rez;
    end
```

- 2) Implementați un algoritm care returnează inversul unui număr.

De exemplu, inversul lui 2345 este 5432.

Răspuns:

Varianta C++:

```
int inversul(int n){
    int init = 0;
    while (n != 0){
        init = init * 10 + n % 10;
        n = n/10;
    }
    return init;
}
```

Varianta Pascal:

```
function inversul(n:integer):integer;
var
init: integer;
begin
    init := 0;
    while (n <> 0) do
    begin
        init := init * 10 + n mod 10;
        n := n div 10;
    end;
    inversul := init;
end;
```

- 3) Considerând că n este un număr natural nenul, explicați care este rezultatul următorului algoritm:

Varianta C++:

```
int ceFace_1(int n){
    int a = 0;
    int b = 1;
    for (int i = 1; i <= n;i++) {
        b = b * i;
        a = a + b;
    }
    return a;
}
```

Varianta Pascal:

```
function ceFace_1(n:integer):integer;
var
a,b,i: integer;
begin
    a := 0;
    b := 1;
    for i := 1 to n do
    begin
        b := b * i;
        a := a + b;
    end;
    ceFace_1 := a;
end;
```

Răspuns:

Funcția `ceFace_1` calculează suma $\sum_{k=1}^n k! = 1! + 2! + \dots + n!$

- 4) Considerând că n este un număr natural, explicați care este rezultatul următorului algoritmul:

Varianta C++:

```
int ceFace_2(int n){
    if (n/10 == 0) {
        if (n % 2 == 0) return 1;
        else return -1;
    }
    return ceFace_2(n/10) + ceFace_2(n%10);
}
```

Varianta Pascal:

```
function ceFace_2(n:integer):integer;
begin
    if (n div 10 = 0) then
        if (n mod 2 = 0) then
            ceFace_2 := 1
        else
            ceFace_2 := -1
    else
        ceFace_2 := ceFace_2(n div 10) + ceFace_2(n mod 10);
end;
```

Răspuns:

Funcția `ceFace_2` calculează diferența dintre numărul de cifre pare și numărul de cifre impare ale lui n .

- 5) Considerând că x și n sunt un numere naturale nenule, cu $x \leq n$, explicați care este rezultatul următorului algoritmul:

Varianta C++:

```
int ceFace_3(int n, int x){
    int a = 1;
    int b = 1;
    for (int i = 0; i < n; i++){
        a = a * (i+1);
        if (i < n - x) b = b * (i+1);
    }
    for (int i = 0; i < x; i++){
        b = b * (i+1);
    }
    return a/b;
}
```

Varianta Pascal:

```
function ceFace_3(n, x:integer):integer;
var
i,a,b,f: integer;

begin
  a := 1;
  b := 1;
  for i := 0 to n do
  begin
    a := a * (i+1);
    if (i < n - x) then
      b := b * (i+1);
    end;
    for i := 0 to x do
    begin
      b := b * (i+1);
    end;
    ceFace_3 := a div b;
  end;
```

Răspuns:

Funcția ceFace_3 calculează expresia $\frac{n!}{x!(n-x)!} = C_n^k$ (combinări de n luate câte k)

6) Implementați un algoritm care să calculeze următoarea sumă:

$$f_2(n) = \sum_{k=1}^n \frac{1}{T_k}$$

unde $T_k = 1 + 2 + \dots + k$

Răspuns:

Varianta C++:

```
double suma_tk(double n){
  if (n == 0){
    return 0;
  }
  return suma_tk(n-1) + 2/(n*(n+1));
}
```

Varianta Pascal:

```
function suma_tk(n: double): double;
begin
  if n = 0 then
    suma_tk := 0
  else
    suma_tk := suma_tk(n-1) + 2/(n*(n+1));
  end;
end;
```

Notă: avem $2/(n*(n+1))$ pentru că $T_k = 1 + 2 + \dots + k = \frac{k(k+1)}{2}$, iar $\frac{1}{T_k} = \frac{1}{\left(\frac{k(k+1)}{2}\right)} = \frac{2}{k(k+1)}$

7) Avem următorul algoritm, unde n este un număr natural, $n > 10$:

Varianta C++:

```
int ceFace_4(int n){
    int a = n % 10;
    n = n/10;
    int b, s = 0, c = 0;
    while (n != 0) {
        b = n % 10;
        int r = 10 * b + a;
        if (r%2 == 0) {
            s += r;
            c++;
        }
        a = b;
        n = n/10;
    }
    return s/c;
}
```

Varianta Pascal:

```
function ceFace_4(n: int64): int64;
var
    a, b, s, c, r: int64;
begin
    a := n mod 10;
    n := n div 10;
    s := 0;
    c := 0;
    while (n <> 0) do
    begin
        b := n mod 10;
        r := 10 * b + a;
        if (r mod 2 = 0) then
        begin
            s := s + r;
            c := c + 1;
        end;
        a := b;
        n := n div 10;
    end;
    ceFace_4 := s div c;
end;
```

Ce va returna algoritmul pentru valoarea 1632012?
(ceFace_4(1632012) = ?)

Răspuns:

Funcția ceFace_4 ia un număr întreg, ia perechile de 2 cifre din acest număr, una după alta și face media celor care sunt pare.

Pantru numărul 1632012 avem numerele de 2 cifre: 12, 01, 20, 32, 63, 16. Dintre acestea sunt pare numerele: 12, 20, 32 și 16. Suma acestora este 80, iar media aritmetică este 20.

Deci pentru 1632012 răspunsul este 20.

8) Avem următorii algoritmi, unde n este un număr natural nenul:

Varianta C++:

```
bool aux_5(int n){
    int i = 0;
    while (i * i <= n) {
        if (i*i == n) return true;
        i++;
    }
    return false;
}
```

```
int ceFace_5(int n){
    int a = 0;
    int b = 10;
    while (b <= n*10){
        int x = n;
        while (x >= b/10) {
            int c = x%b;
            if (aux_5(c)) a++;
            x = x / 10;
        }
        b = b * 10;
    }

    return a;
}
```

Varianta Pascal:

```
function aux_5(n: integer): boolean;
var
    i: integer;
    rez: boolean;
begin
    i := 0;
    rez := false;
    while i * i <= n do
    begin
        if i * i = n then
        begin
            rez := true;
        end;
        i := i + 1;
    end;
    aux_5 := rez;
end;
```

```

function ceFace_5(n: integer): integer;
var
  a, b, x, c: integer;
begin
  a := 0;
  b := 10;
  while b <= n * 10 do
  begin
    x := n;
    while x >= b div 10 do
    begin
      c := x mod b;
      if aux_5(c) then
        a := a + 1;
      x := x div 10;
    end;
    b := b * 10;
  end;
  ceFace_5 := a;
end;

```

Pentru ce număr de 2 cifre algoritmul ceFace_5 returnează valoarea 3?
(ceFace(n) = 3, n = ?)

Răspuns:

Funcția aux_5 verifică dacă un număr este pătrat perfect. Funcția ceFace_5 ia un număr întreg și formează toate numerele posibile din cifre consecutive ale acestui număr, și numără câte dintre ele sunt pătrate perfecte. Spre exemplu, pentru un număr abc, returnează câte sunt pătrate perfecte dintre numerele: c, b, a, bc, ab, abc.

Astfel, un număr de 2 cifre pentru care funcția ceFace_5 returnează 3 este 49 (numerele formate dintre cifrele lui 49 sunt 9 = 3*3, 4 = 2*2 și 49 = 7*7)

9) Implementați un algoritm care să calculeze următoarea sumă:

$$f(n) = \sum_{k=1}^n \frac{1}{k!}$$

Răspuns:

Varianta C++:

```

double suma_k(double n){
  double suma = 0;
  double factorial = 1;
  for (int k =1; k <= n; k++){
    factorial = factorial * k;
    suma = suma + 1/factorial;
  }
  return suma;
}

```

Varianta Pascal:

```
function suma_k(n: integer): double;
var
    suma, factorial: double;
    k: integer;
begin
    suma := 0;
    factorial := 1;
    for k := 1 to n do
        begin
            factorial := factorial * k;
            suma := suma + 1/factorial;
        end;
    suma_k := suma;
end;
```