

METODY PROGRAMOWANIA – LABORATORIUM 3

PROGRAM 01

Wygeneruj plik tekstowy, który zawiera ciąg wszystkich podzbiorów zbioru n-elementowego.

WEJŚCIE

n – ilość elementów zbioru

WYJŚCIE

Plik tekstowy zawierający ciąg wszystkich podzbiorów zbioru n-elementowego, w którym każdy następny podzbiór powstaje z poprzedniego poprzez dodanie lub odjęcie pojedynczego elementu. Każdy podzbiór jest reprezentowany przez ciąg binarny $B[1], \dots, B[n]$.

PSEUDOKOD

```
begin
  for i:=1 to n do B[i]:=0; (* zbiór pusty *)
  i:=0; (* i = liczba dotychczas wygenerowanych podzbiorów *)
  repeat
    write(B[1],...,B[n]);
    i:=i+1; p:=1; j:=i;
    while j mod 2 = 0 do
      begin (*  $j2^{p-1}=i$  *)
        j:= j/2; p:=p+1;
      end; (*  $p = Q(i)+1$  *)
      if p <= n then B[p]:=1-B[p] (* zmiana na pozycji p *)
    until p > n
  end
```

Uwaga! Podziel program na funkcje, pamiętaj o komentarzach do kodu.

PROGRAM 02

Wygeneruj plik tekstowy, który zawiera wszystkie k-elementowe podzbiory zbioru $\{1, \dots, n\}$ w porządku leksykograficznym.

WEJŚCIE

n – ilość elementów zbioru

k – ilość elementów w podzbiorze

k < n

WYJŚCIE

Plik tekstowy zawierający ciąg wszystkich podzbiorów k-elementów zbioru $\{1, \dots, n\}$ w porządku leksykograficznym.

PSEUDOKOD

```
begin
  for i:=1 to k do A[i]:=i; (* pierwszy podzbiór *)
  p:=k;
  while p >= 1 do
    begin write(A[1],...,A[k]);
      if A[k]= n then p:=p-1; else p:=k;
      if p>=1 then
        for i:=k downto p do A[i]:= A[p]+i-p+1;
      end
    end
  end
```

Uwaga! Podziel program na funkcje, pamiętaj o komentarzach do kodu.

PROGRAM 03

Wygeneruj plik tekstowy, który zawiera wszystkie podziały liczby n.

WEJŚCIE

n – liczba

WYJŚCIE

Plik tekstowy zawierający ciąg podziałów liczby n w porządku odwrotnym do leksykograficznego.

PSEUDOKOD

```
begin
  S[1]:=n; R[1]:=1; d:=1; (* pierwszy podział *)
  wypisz podział;
  while S[1] > 1 do (* znajdź następny podział *)
    begin sum:=0; (* sum = suma usuniętych składników *)
      if S[d]=1 then (* usuń składniki równe jedności *)
        begin sum:=sum+R[d]; d:=d-1;
        end;
      sum:=sum+S[d]; R[d]:=R[d]-1; l:=S[d]-1;
      if R[d] > 0 then d:=d+1;
      S[d]:=1; R[d]:=sum div l;
      l := sum mod l;
      if l<>0 then (* dodaj ostatni składnik równy l *)
        begin d:=d+1; S[d]:=1; R[d]:=1;
        end;
      wypisz podział
    end
  end
```

end

ZASADY ODDAWANIA GOTOWYCH PROGRAMÓW:

Pliki **.cpp** o nazwach:

Nazwisko_Imie_Program_01.cpp

Nazwisko_Imie_Program_02.cpp

Nazwisko_Imie_Program_03.cpp

oraz wszystkie inne utworzone pliki w tym pliki tekstowe zawierające wyniki powinny być zamieszczone w katalogu: **Nazwisko_Imie_Laboratorium_3**

Katalog powinien być spakowany w formacie **.rar** lub **.zip** i przesłany do folderu: **Programy - laboratorium 3 – Poniedziałek godzina 19.45** dostępnego na stronie kursu MP (elf2.pk.edu.pl)

LITERATURA:

Balińska Krystyna, Zwierzyński Krzysztof: „Projektowanie algorytmów grafowych”, Wydawnictwo Politechniki Poznańskiej, 2002.

Lipski Witold: „Kombinatoryka dla programistów”, Wydawnictwo Naukowo Techniczne, 2004.