PROGR. RÓWN. WSPÓŁB. I ROZPR.

Laboratorium 2

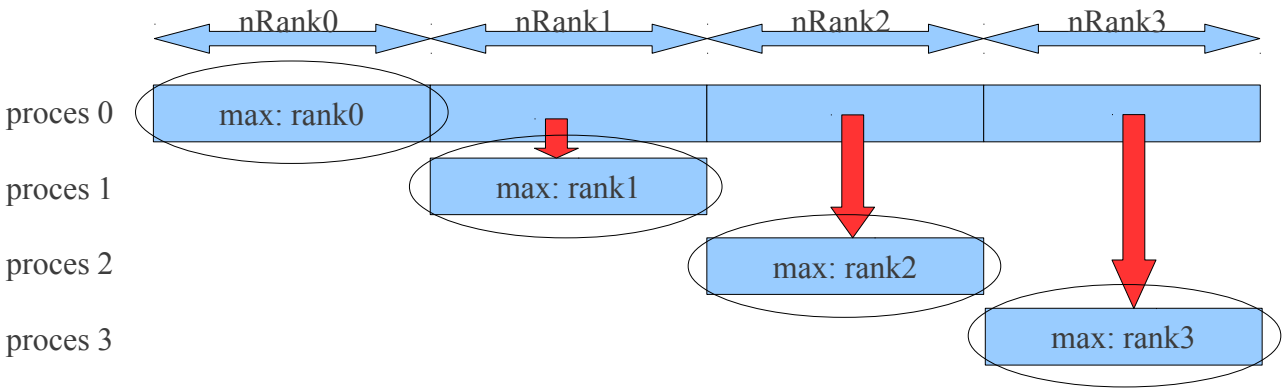
Wykorzystując bibliotekę MPI napisać równoległą wersję poniższego kodu:

```
float max(const float &a, const float &b) {
  return (a>b)?a:b;
}

int main(int argv, char **argc) {

  int n;
  std::cout<<"\nwprowadz rozmiar tablicy:\n";
  std::cin>>n;

  int * tab = new int[n];

  for(int i = 0; i<n; ++i) {
    tab[i] = i;
  }

  int mx;
  for(int i = 1; i<n; ++i) {
    mx = max(mx, tab[i]);
  }

  std::cout<<"max: "<<mx<<std::endl;
  delete[]tab;
}
```

Założenia:

- dowolny rozmiar tablicy n;
- dowolna liczba procesów (n>>size);
- podstawowe zadania dla procesu 0:
  - dynamicznie przydzielić pamięć dla głównej tablicy *tab* o rozmiarze n,
  - zapewnić równomierny podział obliczeń pomiędzy dostępne procesy (rys.): wyznaczyć rozmiar zadania *nRank* dla każdego procesu,
  - do każdego procesu wysłać rozmiar poszczególnych fragmentów głównej tablicy (*nRank*),
  - przesłać odpowiednie fragmenty głównej tablicy do każdego procesu,
- podstawowe zadania dla procesów: od 1 do (size-1):
  - odebrać rozmiar *nRank* poszczególnych części tablicy,
  - dynamicznie przydzielić pamięć dla tablicy o rozmiarze *nRank*,
  - odebrać odpowiedni fragment tablicy,
- podstawowe zadania dla wszystkich procesów:
  - znaleźć maksymalny element *maxRank* w obrębie każdego procesu,
  - wykorzystać funkcję MPI_Reduce do znalezienia maksymalnego elementu *maxAll* spośród znalezionych elementów w poszczególnych procesach
    MPI_Reduce (*sendbuf, *recvbuf, count, datatype, op, root, comm)
    MPI_Reduce(&maxRank, &maxAll, 1, MPI_INT, MPI_MAX, 0, MPI_COMM_WORLD)
  - zwolnić obszar pamięci dynamicznej;

Uwaga!
W celu uproszczenia można przyjąć: **n=100** oraz **-np 4**



Wskazówki:

MPI: http://icis.pcz.pl/~roman/mpi-www/

Podstawowe polecenia:
lamboot, lamclean, mpic++, mpirun

Kompilacja:
mpic++ -o lab ./lab.cpp

Uruchamiania:
mpirun -np **x** ./lab
gdzie **x** to liczba procesów