**Текстовая обработка данных**

**var**

key:char;

a,i,j:integer;

s:string;

**begin**

readln(s); //считываем слово для кодирования

**for** i := 1 **to** length(s) **do begin** //цикл для перебора каждого введенного символа

a:=OrdAnsi(s[i]); //присваиваем в переменную а код ANSI (в правильном порядке)

**for** j:= 1 **to** 3 **do** //цикл для "переворачивания" кода символа

**begin**

write(a **mod** 10); //отдельно выводим каждую цифру из числа в обратном порядке

a:=a **div** 10; // убирая у числа a, выведенную цифру

**end**;

write(' ');

**end**;

**end**.

//Function **Length**(A : DynArrayType): Integer; Функция возвращает длину строки **S**

**//**только это числа не по ASCII, а по ANSI. потому что в PascalABC используется именно эта кодировка

только это числа не по ASCII, а по ANSI. потому что в PascalABC используется именно эта кодировка

ну и если что, "чтоб было по ASCII надо добавлять большой словарь соответствий вручную"

просто задание скорее всего для других паскалей, где кодировка ASCII была

Динамические структуры

{8. Справочный зал имеет M терминалов. Поток пользователей равномерно распределен во времени.

Среднее количество пользователей равно P чел/сутки. Время работы одного пользователя на терминале

составляет в среднем t минут. Определить, среднее время ожидания.

(Например, m=5, p=140 чел/сутки, t=40 мин, Т0=24 часа).}

**type** data = **record**

number, next, T\_ochered: integer;

**end**;

**const** p = 7;//поток

//const M = 4;//количествo терминалов

**const** t = 60;//время работы одного пользователя

**const** T0 = 1440;//общее время

**const** kolvo\_polz = T0 **div** p + 1;//количество пользовтелей, которое успеют обслужить за общее время

**Var** ochered: **array** [0..kolvo\_polz] **of** data;//массив, в котором будут содержаться данные по каждому пользователю(его номер, какой номер будет на этом терминале после него, время стояния в очереди)

count: integer;

terminal\_t1, terminal\_t2, terminal\_t3, terminal\_t4: integer;//счетчик времени обслуживания на каждом из терминалов

tail: integer;//в переменной хранится номер пользователя, которое стоит в конце очереди

summa\_T\_ochered: integer;//общее время стояния в очереди

razmer\_ocheredi: integer;// размер очереди в данное время

i, j: integer;

average\_T\_ochered: real;//среднее время стояния в очереди

terminal\_1\_free, terminal\_2\_free, terminal\_3\_free, terminal\_4\_free, k: boolean;//переменные, по которым можно будет узнать, свободен ли терминал

{добавление в очередь}

**Procedure** AddToOchered(**var** count:integer);

**begin**

ochered[count].number := count;//заполняем данные про пользователя

ochered[count].next := 0;

ochered[count].T\_ochered := 0;

**if** count <> 1 **then**

ochered[count - 1].next := count+2

**else**

ochered[count - 1].next := count;

tail := count;

**end**;

{сдвиг очереди}

**Procedure** SdvigOcheredi();

**begin**

ochered[0].next += 1;//показывает, с каким номером пользователь в очереди после текущего

**end**;

**begin**

ochered[0].number := 0;

ochered[0].next := 1;

ochered[0].T\_ochered := 0;

tail := 0;

terminal\_1\_free := True; terminal\_2\_free := True; terminal\_3\_free := True; terminal\_4\_free := True;

count := 0;

k := False;

summa\_T\_ochered := 0;

**for** i:=1 **to** T0 **do begin**

**if** p > 1 **then begin**

**if** i **mod** p = 1 **then begin**//через каждые p минут добавляем в очередь пользователя

count += 1;

k := True;//доп переменная указывает на то, что пришел новый пользователь

AddToOchered(count);

**end**;

**end**

**else**

**if** p = 1 **then begin**

count += 1;

k := True;

AddToOchered(count);

**end**;

{если на каком-то из 4 терминалов пользователя обслужили полностью, то есть счетчик времени достиг времени обслуживания, то

этот терминал становится свободным, а счетчик времени на нем обнуляется}

**if** terminal\_t1 = t **then begin**

terminal\_1\_free := True;

terminal\_t1 := 0;

**end**;

**if** terminal\_t2 = t **then begin**

terminal\_2\_free := True;

terminal\_t2 := 0;

**end**;

**if** terminal\_t3 = t **then begin**

terminal\_3\_free := True;

terminal\_t3 := 0;

**end**;

**if** terminal\_t4 = t **then begin**

terminal\_4\_free := True;

terminal\_t4 := 0;

**end**;

{Если какой-то из 4 терминалов занят, значит, на нем еще обслуживается пользователь, поэтому к счетчику времени на этом терминале добавляем +1}

**if** terminal\_1\_free = False **then**

terminal\_t1 += 1;

**if** terminal\_2\_free = False **then**

terminal\_t2 += 1;

**if** terminal\_3\_free = False **then**

terminal\_t3 += 1;

**if** terminal\_4\_free = False **then**

terminal\_t4 += 1;

{Если какой-то из 4 терминалов свободен и дополнительная логическая переменная равна истине(то есть пришел новый пользователь), то очередь сдвигается,

пользователь занимает свободный терминал, и он становится занятым.

Счетчик времени равен 1, так как началось обслуживание. k становится равным Ложь,

чтобы остальные терминалы тоже не сделать занятыми}

**if** terminal\_1\_free **and** k **then begin**

Sdvigocheredi;

**if** razmer\_ocheredi > 0 **then**

razmer\_ocheredi -= 1;//как только пользователь из очереди занимает свободный терминал, размер очереди уменьшается на 1

terminal\_1\_free := False;

terminal\_t1 := 1;

k:=False;

**end**

**else**

**if** terminal\_2\_free **and** k **then begin**

Sdvigocheredi;

**if** razmer\_ocheredi > 0 **then**

razmer\_ocheredi -= 1;

terminal\_2\_free := False;

terminal\_t2 := 1;

k:=False;

**end**

**else**

**if** terminal\_3\_free **and** k **then begin**

Sdvigocheredi;

**if** razmer\_ocheredi > 0 **then**

razmer\_ocheredi -= 1;

terminal\_3\_free := False;

terminal\_t3 := 1;

k:=False;

**end**

**else**

**if** terminal\_4\_free **and** k **then begin**

Sdvigocheredi;

**if** razmer\_ocheredi > 0 **then**

razmer\_ocheredi -= 1;

terminal\_4\_free := False;

terminal\_t4 := 1;

k:=False;

**end**;

{считаем, сколько времени пользователь стоит в очереди}

**for** j := ochered[0].next **to** tail **do begin**

**if** ochered[j].T\_ochered = 0 **then**

razmer\_ocheredi += 1;

ochered[j].T\_ochered += 1;

**end**

**end**;

{после выхода из цикла for считаем общее время стояния в очереди всех пользователей}

**for** j := 1 **to** tail **do**

summa\_T\_ochered += ochered[j].T\_ochered;

{находим среднее время стояния в очереди. делим общее время стояния в очереди на количество пользователей}

average\_T\_ochered := summa\_T\_ochered / (count - razmer\_ocheredi);

writeln('Общее время в очереди ', summa\_T\_ochered, ' минут');

writeln('Среднее время стояния в очереди ', average\_T\_ochered, ' минуты');

**if** razmer\_ocheredi > 0 **then begin**

write('Через ', T0, ' минут в очереди будут стоять ', razmer\_ocheredi, ' пользователей c номерами ');

**for** i:=razmer\_ocheredi **downto** 1 **do**

write(ochered[kolvo\_polz - i].number, ' ');

writeln;

**end**

**else**

**if** razmer\_ocheredi = 0 **then**

write('Через ', T0, ' минут в очереди пользователей не будет');

**end**.

ну типа берется массив вот этот, с очередью. и по очереди засовывается в свободные терминалы, по истечению времени терминал особождается - засовывается новый пользователь в него

время общее заканчивается - все финиш. показывает какие пользователи остались. подсчитывает сколько времени всего надо и тд

12:02