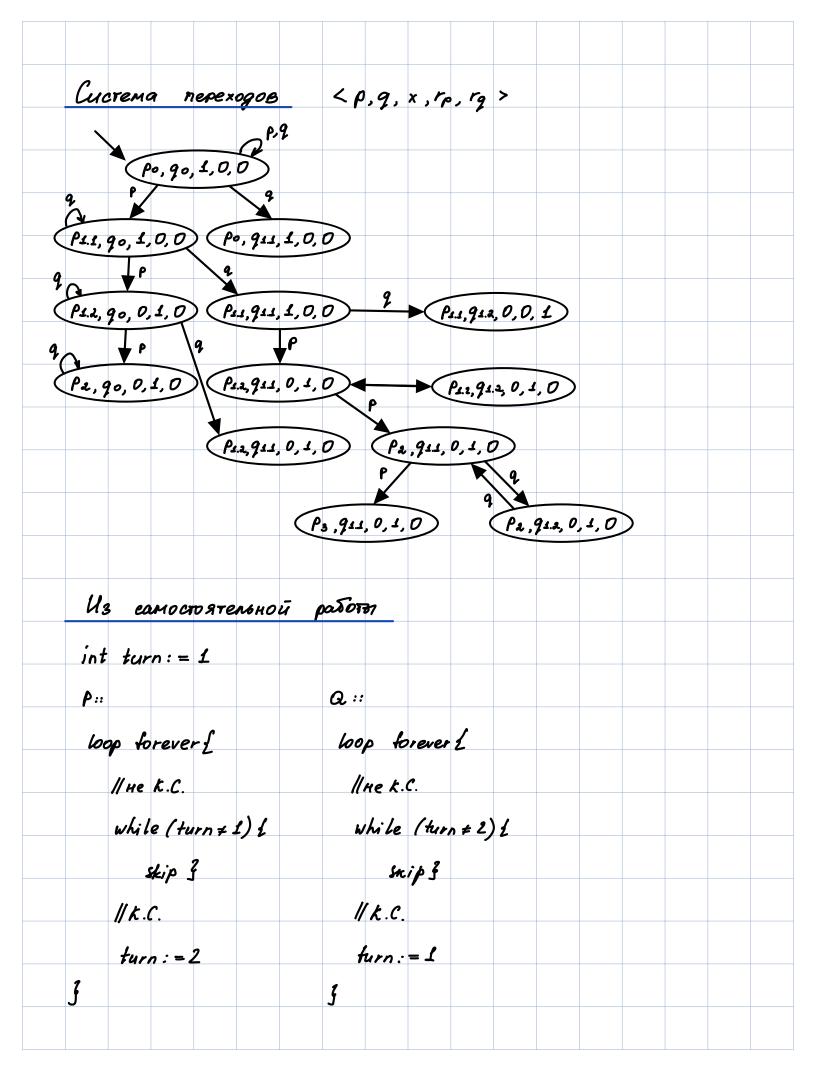
Nex.	ция	8.	Pas	50P	30	gau	KI	u l	?P							
										wuje	ero	90	ступ	a c	c n	0 -
Į,	ใกя	<i>э</i> той	80	igay	u	noca	POUT	3 (en,	на	неи	1 0	10 5 P A	3U72	H	епрс
ecc 4	ВНУН	0 H	екри	ruze	CKYH	, ce	zcyu	ю	u n	POPPE	CCUBI	491-0	K.C	•		
x -	Perl	ICTP	×	:=1	•											
acqu	uire -	mut	ex()	£						com	pare	and	- sw	ар (о	d, neu	~) {
	boo	l r	-/	oka/	16 HQ 8	Nep	енен	ная			;f	(x =	old)	٤		
	·											_				
				mpa	re_	and.	. swaj	o (1,	0)		Z	retu	irn (7	true)		
•	Junt	il (r)								J +.			\		
ea la	000	 +	- ()							ż	1644	PN /1	-a/se			
			ex ()	L												
Pea	nusa	ция														
, 00p	for	rever	1													
//	/ не	k.C.														
	acqu	ire_	mut	ex()												
_//	K.C	•														
r	eleas	e_n	nute	()												
	Deal Deal	Peuniousho Ang eccushyn x - period acquire - boo repea y unt elease - x:=. Deanusa oop for line acquire.	Pewurb Duyblo com Ang stoid Eccubition in K-peructa Acquire_mut bool r repeat { repeat {	Peulut saga ougho compare Ang stou sa eccubhyo hexpu x - peructp x acquire mutex() bool r - n repeat { r:= x. ce y until (r) elease mutex () x:= 1 Deanusayus oop forever { lhe k.C. acquire mutex ()	Pewath sagays DULLED COMPARE_AN ANA STOU BAGAY ECCUBHYLO HEXPUTURE (- PERUCTP X:= 1 Acquire_mutex(){ bool r - noxal repeat { repeat { x:= 1 Peanusayus Poop forever { ll He k.C. acquire_mutex() acquire_mutex()	Peunto sagayy 63. Duy blo compare_and_sw Lng stou sagayy Eccubhyo Hexputuzeckyh K-Peructt x:=1 Acquire_mutex(){ bool r -nokanbhag repeat { r:= X.compare_ Juntil (r) elease_mutex(){ x:=1 Peanusayus Dop forever { ll he k.C. acquire_mutex() ll K.C.	Peuluth sagays BSAUME DULLED COMPARE and SWAP LINA STOUT BAGAYU PROCES EXCUBHYD HEXPUTUZECKYHO CE K-PERUCTP X:=1 Acquire mutex(){ bool r -noxanbhas nex repeat { r:= X. compare and. Juntil (r) elease mutex () { x:= 1 Peanusayus pop forever { ll he k.C. acquire mutex() ll K.C.	Peuurb задачь взаинно ощью compare_and_swap (old, Для этой вадачи гостроит ессивнью некрититескую секци. «- регистр х:=1 ведите_mutex(): bool r - локальная перенен repeat [r:= x. compare_and_swap Juntil (r) elease_mutex(): x:=1 Peanusayus pop forever { // не k.C. acquire_mutex() // К.C.	Pecuur задачь взаимно исклодью сотрате_and_swap (old, new Ina этой вадачи построить (остроить и сесивным некритическую секцию секцию секцию и к-репистр х:=1 Require_mutex(){ bool r - покальная переменная repeat { r:= x. compare_and_swap(1,	DULLE COMPARE AND SWAP (old, new) In a stou sagayu rocerpoute CN Reccussion Hexputure cxy to cexy upo u n K - peructe x:=1 Acquire mutex() { bool r - nokane has nepenenhas repeat { r:= X. compare and swap (1,0) Juntil (r) elease mutex () { x:= 1 Peanusayus pop forever { ll he k.C. acquire mutex() ll K.C.	Pewurb sagano Bernho ucknowaringer Duybo compare_and_swap (old, new) Ling stow sagany recorpouts CN, ha eccubhyo hexputuzeckyho cekyuro u reporter K - peryctp x:=1 require_mutex(){ bool r - nokanbhar nepenenhar repeat { r:= X. compare_and_swap(1,0) Juntil (r) elease_mutex(){ x:=1 Peanusayur pop forever { line k.C. acquire_mutex() lik.C.	Pewuth задачи взаинно исключающего ощью compare_and_swap (old, new) Для этой вадачи гостроить СП, на нее ессивную некритическую секцию и прогрессива (- регистр x:=1 пеquire_mutex() { ri= X. compare_and_swap(1,0) Juntil(r) Juntil(r) selease_mutex() { x:=1 Peanusayus pop forever { // не k.C. acquire_mutex() // K.C.	Pemure sagayu взаимно исключающего до ощью compare_and_swap (old, new) Для этой вадачи построить СП, на ней ого ессивную некрититескую секцию и прогрессивную к - регистр x:=1 пеquire_mutex(): bool r - покальная переменная repeat! r:= x. compare_and_swap(1,0) yuntil(r) selease_mutex(): x:=1 Peanusayus pop forever! // He k.C. acquire_mutex() // K.C.	Pewurb задачь взаинно исключающего доступ ощью compare_and_swap (old, new) Для этой вадачи построить СП, на ней отобра ессивную некрититескую секцию и прогрессивную к.С к - регистр x := 1 ведиіге_mutex() { bool r - покальная перененная r := x. compare_and_swap(1,0) funtil(r) guntil(r) return(false) elease_mutex() { x := 1 Peanusayus pop forever { l/ не k.C. acquire_mutex() l/ k.C.	Peuus задачь взеинно исключающего доступа опицью сотрате_and_swap (old, new) Для этой задачи построить СП, на ней отобразить сесивную мекритигескую секцию и прогрессивную к.С. (- регистр x:=1 подшіге_mutex() { repeat { r:= x. compare_and_swap(1,0) } until (r) } elease_mutex() { x:= 1 Peanusayux pop forever {	Pewurs задачь взаинно исключающего доступа с поривно сотрате_and_swap (old, new) Для этой задачи построить СП, на ней отобразить не прешенью нехрититескую секцию и прогрессивную к.С. К - регистр х:=1 require_mutex() f bool r - покальная перененная if (x=old) f repeat f r:= x. compare_ and_swap (f, 0) Juntil (r) selease_mutex () f x:=1 Peanusayur pop forever f // не к.С. acquire_mutex() // к.С.



f	la I	Prome!	la					<u>l</u>	<u>.</u> Τ <u>.</u> ,							
in	it t	urn =	1						[] (v	vant	[0] -	→ <>	criti	[0]		
bo	ool	crit[2], w	ant I	2] =	{o, 0	o }						crit			
a	ctive	[2]	procts	pe P	() {				7 7	(crit	[0]	& crit	£ [±])			
	lo ::															
	ſ	do														
ene		:: skip	o													
	4	:: bre														
		d														
	NO	mt[_	pid I :	- 1												
		ant[•													
		(turn														
	cr	it[_	pid]=	1												
		it I_														
		rn = 1														
00			,													
3																
An	ropu	m I	ererce	OHA	<i>g</i> 1 8	2	пр	oyecc	OB							
				IR a	-					7						
				RNII						\	Реиме	новал	и			
6	bool	FLA	G [o]	· = 0	, FL	AG [<u> 17:</u>	= O ,	AFT	ER_	you	:= i		, j e	{0,1	}
			WMR		•	SWM				MW	MR			<i>y</i> .	-	-

aca	uire	_ mi	dex i	(;) {											
7				., L											
	FLA	G[i]	=1												
	AF	TER_	.904	:= i											
	whi	le (FLAC	G [j]	7 = 1	2 AI	FER	_904	(= i)	{sk	ip }				
3															
re la	ase_	mut	ev (i	ۍ (
_	Flag]'].	=0												
}															
Peu	ાભાપ	e	n-n	роиз	води	itene	ū	u n	n-no	теба	1Теле	zū	C M	ОНИТ	орам
	прави														
mo	ni tor														
	int	k :=	m + .	n											
	NB_	FUL	(, : = (9											
	BUF	[0	. k - 1	7											
	cond	d C	- PRC	D (C_ CC	DNS,	C-M	UTE	X						
7				,				\				7			
J									rako	е	genoe	ue ?	ero	нет.	
Β.ρ	rodu														
	<i>;</i>	(NB	_FU	LL =	k) :										
		C_PI	ROD.	wait	()										
	z														
					()		ecn	u ni	ouec	c n	onan	CHO	ga.	TD	

	BUF [in]. write (1)	
	in:= (in+1) mod k	
	C-MUTEX. signal()	
	NB_FULL:=NB_FULL +1	
	C. CONS. signal()	
3		
В.	concume (i) {	
	if (NB-FULL = 0) {	
	C_CONS. wait()	
	7	
	C_MUTEX.wait()	
	C-HUIEX.WAIL	
•		
ß		