

Subiectul 1

1. EXPRESIE

Scrieti un program in limbaj de asamblare care calculeaza rezultatul expresiei fara a folosi instructiunile coprocesorului matematic.

Sa se implementeze o procedura care calculeaza urmatoarea expresie:

$$f(x,y,z)=x^2 - \frac{3*(y+1)-\frac{z}{5}}{4}$$

In procedura se implementeaza doar calculul expresiei!

Valorile parametrilor (x,y,z) sunt numere **intregi pozitive**, introduse de la tastatura.

Rezultatul o sa fie afisat pe ecran.

Punctaje:

- Citirea numerelor de la tastatura – 0.1
- Calculul corect al expresiei -0.4
- Afisare in consola -0.1
- Implementarea si apelarea corecta a procedurii - 0.3
- Comentarea codului – 0.1

Exemple verificare:

x=3 y=4 z=15 => f(x,y,z) = 6

x=2 y=7 z=20 => f(x,y,z) = -1

2. CODIFICARE

Se citeste un sir de caractere dintr-un fisier text, in.txt; si se mai citeste un caracter de la tastatura.

In fisierul out.txt se va gasi sirul respectiv codificat. Pentru codificare, fiecare caracter o sa fie inlocuit de caracterul cu 3 pozitii mai la dreapta, iar la scrierea in fisier, daca in rezultatul codificarii se intaleste caracterul citit, el nu va fi afisat.

De exemplu: caracterul citit este f

cde->gh (c se codifica in f care nu se afiseaza)

colocviu -> rorylx

Punctaje:

- Citirea din fisier - 0.2
- Implementarea corecta a algoritmului de codificare intr-un macro care calculeaza in EAX valoarea codificata a caracterului specificat prin parametrul metodei.-0.2
- Apelarea macroului pentru fiecare caracter din sir – 0.2
- Scrierea in fisier respectand inversarea caracterelor doua cate doua-0.2
- Inchiderea fisierelor – 0.1
- Comentarea codului – 0.1

Subiectul 2

1. EXPRESIE

Scrieti un program in limbaj de asamblare care calculeaza rezultatul expresiei fara a folosi instructiunile coprocesorului matematic.

Sa se implementeze o procedura care calculeaza urmatoarea expresie:

$$f(x,y,z)=\frac{1}{2} * (x + 1) + \frac{y+\frac{y}{2}}{4-z}$$

In procedura se implementeaza doar calculul expresiei!

Valorile parametrilor (x,y,z) sunt numere **intregi pozitive**, introduse de la tastatura.

Rezultatul o sa fie afisat pe ecran.

Punctaje:

- Citirea numerelor de la tastatura – 0.1
- Calculul corect al expresiei -0.4
- Afisare in consola -0.1
- Implementarea si apelarea corecta a procedurii - 0.3
- Comentarea codului – 0.1

Exemple verificare:

x=3 y=4 z=2 => f(x,y,z) = 5

x=5 y=10 z=9 => f(x,y,z) = 0

2. CODIFICARE

Se citeste un sir de caractere dintr-un fisier text, in.txt; si se mai citeste un caracter de la tastatura. In fisierul out.txt se va gasi sirul respectiv codificat. Pentru codificare, fiecare caracter o sa fie inlocuit de caracterul cu 2 pozitii mai la stanga, iar la scrierea in fisier, daca in rezultatul codificarii se intaleste caracterul citit, el va fi dublat.

De exemplu: caracterul citit este a

c->aa, 3d->1b

Colocviu -> Amjmtgs

Punctaje:

- Citirea din fisier - 0.2
- Implementarea corecta a algoritmului de codificare intr-o procedura care returneaza valoarea codificata a caracterului specificat prin parametrul metodei. Procedura trebuie sa respecte conventia **fastcall** -0.2
- Apelarea acestei proceduri pentru fiecare caracter din sir – 0.2
- Scrierea in fisier respectand inversarea caracterelor doua cate doua -0.2
- Inchiderea fisierelor – 0.1
- Comentarea codului – 0.1

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]