

Disciplina	Data de Entrega das Soluções
IAL-002 – Algoritmos e Lógica de Programação	Verificar no site do professor e na tarefa do Teams
Nome deste Projeto Programa e que deve ser exibido na tela na execução do mesmo	
Gerador de Senhas	

Requisitos para entrega

1. Atividade em grupo de 3 ou 4 alunos. Entregas individuais podem, eventualmente, ser aceitas com autorização prévia do professor (situação de exceção para os alunos que não sejam da turma).
2. Os programas devem ser entregues através da tarefa do Teams "Atividade de Avaliação N2.B", respeitando o prazo estipulado.
3. É necessário que apenas um integrante do grupo faça o upload das soluções no Teams.
4. Cada projeto programa deve estar em um arquivo e os nomes dos arquivos estão indicados em vermelho no título do projeto.
5. Cada programa deverá exibir na tela, no seu início, o nome do projeto programa e o nome completo dos integrantes do grupo. **Haverá o desconto de 2,5 pontos se isso não for feito.** Isso é necessário para minha organização, pois no final do semestre tem muita coisa para corrigir e atribuir nota, de modo que se os dados estiverem na tela fica tudo melhor e mais rápido. Conto com a colaboração de vocês.
6. Os programas devem ser escritos em Linguagem Python 3. Podem usar as funções de listas que desejarem.
7. Bibliotecas externas não podem ser usadas, com exceção da biblioteca *random*, se necessário.
8. Programas iguais entregues por diferentes grupos será considerado plágio e todos os envolvidos terão as notas zeradas.

Enunciado: Gerador de Senhas – arquivo N2B.py

Descrição do Projeto

Este projeto estará baseado na leitura de um arquivo texto de entrada contendo números de matrícula de alunos de uma escola. Sabe-se que cada nº de matrícula tem 6 dígitos. O arquivo de entrada deve ter o nome MATR.TXT e para realizar os testes do programa vocês devem criá-lo usando um editor de texto simples (bloco de notas, por exemplo). Para cada nº de matrícula presente no arquivo deve ser gerada uma senha conforme as condições especificadas abaixo. Tanto o nº de matrícula como a senha gerada devem ser gravados no arquivo de saída SENHAS.TXT com o formato a seguir:

Exemplo: neste exemplo foram geradas senhas do tipo Alfanumérica 1, com 7 caracteres

MATR . TXT

330019
414061
109229
827392
etc...

SENHAS . TXT

330019;318A89P;
414061;E87H14M;
109229;019MKX9;
827392;313G093;

Condições para geração de senhas

No início do programa, antes de efetuar a leitura do arquivo de entrada, o programa deve pedir que o usuário informe:

- O tipo de senha:
 - Númerica – conterá apenas algarismos;
 - Alfabética – conterá apenas letras maiúsculas e minúsculas;
 - Alfanumérica 1 – conterá letras maiúsculas e algarismos;
 - Alfanumérica 2 – conterá letras maiúsculas, minúsculas e algarismos;
 - Geral – conterá letras maiúsculas, minúsculas, algarismos e os caracteres "-", "_", ":", "@", "#", "\$", "&", "?"
- O tamanho da senha – quantidade de caracteres que a mesma deve conter

Requisito importante que fará parte da avaliação

A senha deve ser gerada dentro de uma função denominada GeraSenha. Essa função receberá dois parâmetros, conforme a escolha do usuário para as duas informações descritas acima:

- Tipo: que é um caractere que deverá ser 'a', 'b', 'c', 'd', 'e' e definirá o tipo de senha gerada
- Tam: que é o tamanho da senha, um número inteiro

O retorno da função é um string contendo a senha gerada. Assim, o cabeçalho e o retorno da função devem ser:

```
def GeraSenha(Tipo, Tam):  
    # aqui virá todo o código necessário para gerar a senha  
    return senha
```

Tabela ASCII

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]