

Disciplina	Turma	Nome do Projeto	Data de Entrega
IAL002 – Algoritmos e Lógica de Programação	Noite	N2.B – Senhas – 2º Sem. 2024	Ver na tarefa Teams

Instruções:

- Leia e cumpra atentamente todos os itens dessas instruções;
- Na resolução deste Projeto-Programa é obrigatório usar a linguagem Python;
- Esta é uma atividade em grupo. Cada grupo poderá ter até 4 alunos. Não serão aceitos grupos maiores. Entregas em duplas ou individuais podem, eventualmente, ser aceitas com autorização prévia do professor (principalmente em casos de grupos cujos alunos tenham desistido da disciplina ou nos casos de dependência não presencial);
- A entrega deverá ser feita exclusivamente através da tarefa do Teams criada para esse fim. Qualquer outro meio de envio usado sem autorização do professor será ignorado;
- Apenas um integrante do grupo deverá fazer a entrega. Os demais integrantes do grupo não devem fazer nada no Teams;
- Fiquem atentos ao prazo de entrega. Esse prazo está estabelecido na tarefa do Teams. Entregas após o prazo/horário não serão aceitas;
- Nas primeiras linhas do código fonte deve haver o nome completo de cada aluno do grupo;
- A não entrega deste projeto na data estipulada e no início da aula acarretará nota igual a zero para este trabalho;
- Se for constatado plágio (integral ou parcial) entre os grupos, a nota de todos os grupos envolvidos será zerada;

Descrição do Trabalho

Informações Iniciais

Este projeto estará baseado na leitura de um arquivo texto de entrada contendo números de matrícula com 6 dígitos de alunos de uma escola. Esse arquivo de entrada deve ter o nome **MATR.TXT** e deverá conter um número por linha. Para cada nº 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 indicado no exemplo abaixo.

Para testar o programa vocês devem criar um arquivo de entrada usando o bloco de notas e sugiro que ele tenha pelo menos 10 linhas.

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 dois parâmetros de entrada:

- O tipo de senha:
 - Numérica deverá conter apenas algarismos;
 - Alfabética deverá conter apenas letras maiúsculas e minúsculas;
 - Alfanumérica 1 deverá conter letras maiúsculas e algarismos;
 - Alfanumérica 2 deverá conter letras maiúsculas, minúsculas e algarismos;
 - Geral deverá conter letras maiúsculas, minúsculas, algarismos e caracteres especiais selecionados a partir deste conjunto: +, -, _, !, ?, #, @, *, &, =, \$, %

As letras A, B, C, D, E maiúsculas deverão ser usadas para essa finalidade. Qualquer outro caractere deve gerar na tela a mensagem "Tipo de senha inválido".

- O tamanho da senha Quantidade de caracteres que ela deve conter
O tamanho da senha não pode ser inferior a 6 nem maior que 25 caracteres.

Requisito obrigatório

O programa deve conter uma função chamada `GeraSenha(tipo, tamanho)`. Essa função recebe dois argumentos: a letra para o tipo de senha e um inteiro para o tamanho.

A função deve retornar a senha gerada.

Apenas a função `GeraSenha` é obrigatória, mas caso queiro criar outras funções, fiquem à vontade.

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]

Esta tabela ASCII é fornecida para quem desejar usar a geração de números inteiros e depois convertê-los para caractere usando a função `chr()` do Python.

Para saber o que faz a função `chr()` de Python teste no Idle os comandos abaixo:

```
>>> chr(65)
>>> chr(66)
>>> chr(90)
>>> chr(97)
>>> chr(98)
>>> chr(122)
>>> chr(48)
>>> chr(49)
>>> chr(57)
>>> chr(43)
>>> chr(45)
>>> chr(63)
>>> chr(33)
```