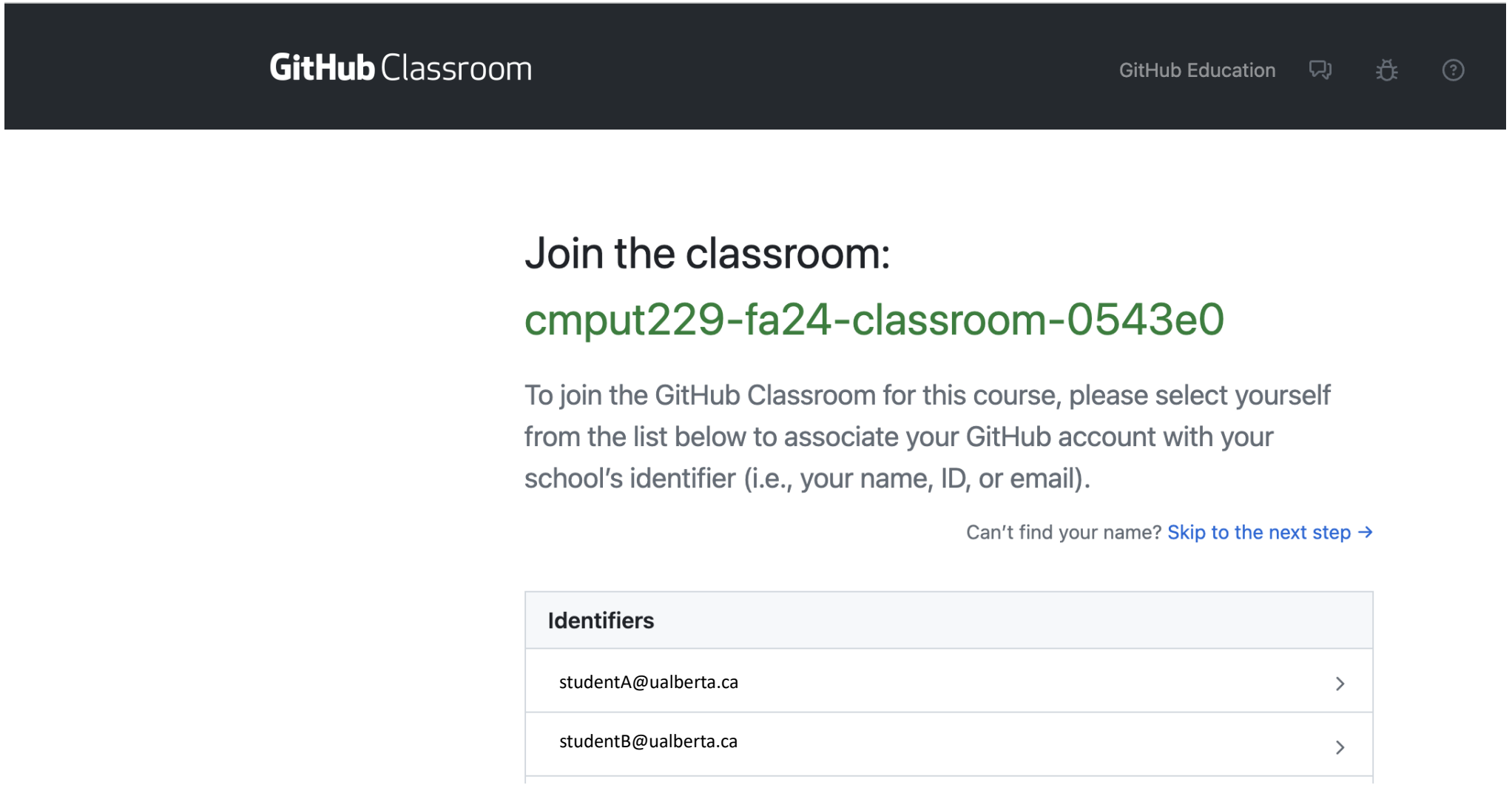
CMPUT 229

Lab #2: String Search

Important things first

- GitHub Classroom
- CMPUT229 Student Submission License
- Private repos

GitHub Classroom



Learn to git

Lab Information

Git and Github

Git and Github are very important version control management systems that are widely used by software practitioners in the industry. The sooner students in this area become familiar with these tools, the more productive and qualified they will be. To encourage students to get started with their learning, the labs in CMPUT229 are distributed through gitclassroom. Students are expected to learn the basics about github on their own. There are very tutorials in different forms on the web. Here is one that we found very useful:

- Kevin Stratvert's Git and GitHub for Beginners Tutorial

This is a short video where Quinn Pham explains how to clone a lab and how to use the command line in a terminal to issue git commands:

- How to clone a lab and how to submit a solution

The git command cheat sheet that Quinn refers to is here:

- Git command cheat sheet

https://cmput229.github.io/229-labs-RISCV/

```
# CMPUT 229 Student Submission License
# Version 1.0
#
# Copyright 2024 <student name>
#
# Redistribution is forbidden in all circumstances. Use of this
# software without explicit authorization from the author or CMPUT 229
# Teaching Staff is prohibited.
#
# This software was produced as a solution for an assignment in the course
# CMPUT 229 - Computer Organization and Architecture I at the University of
# Alberta, Canada. This solution is confidential and remains confidential
# after it is submitted for grading.
#
# Copying any part of this solution without including this copyright notice
# is illegal.
#
# If any portion of this software is included in a solution submitted for
# grading at an educational institution, the submitter will be subject to
# the sanctions for plagiarism at that institution.
#
# If this software is found in any public website or public repository, the
# person finding it is kindly requested to immediately report, including
# the URL or other repository locating information, to the following email
# address:
#
#          cmput229@ualberta.ca
```

This solution is confidential
and remains confidential
After it is submitted for grading.

⇒ Your solution for a CMPUT 229 lab must remain in a PRIVATE git repository.
Even AFTER you graduate from the UofA.

CMPUT 229

Background

How are Strings Represented?

Strings are represented as arrays of characters:

Each character in the string is encoded as a number.

ASCII is a standard that maps between characters and numbers.

Each ASCII character is a single byte, and characters range from 0–127.

String termination:

When passing a string to a function, we only pass a pointer to the first character.

To know when the string ends, we include a terminating character after the last character of the string.

In ASCII the terminating character is 0x00, which is often represented as ‘\0’.

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	*[END 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]*

Example String in Memory

What does the string “This is a test” look like in memory?

We are given a pointer to the first character of the string.

‘T’	‘h’	‘i’	‘s’	‘ ’	‘i’	‘s’	‘ ’	‘a’	‘ ’	‘t’	‘e’	‘s’	‘t’	‘\0’
0x54	0x68	0x69	0x73	0x20	0x69	0x73	0x20	0x61	0x20	0x74	0x65	0x73	0x74	0x00

String Search

String search in this lab:

String search finds the locations of a query string in a search string.

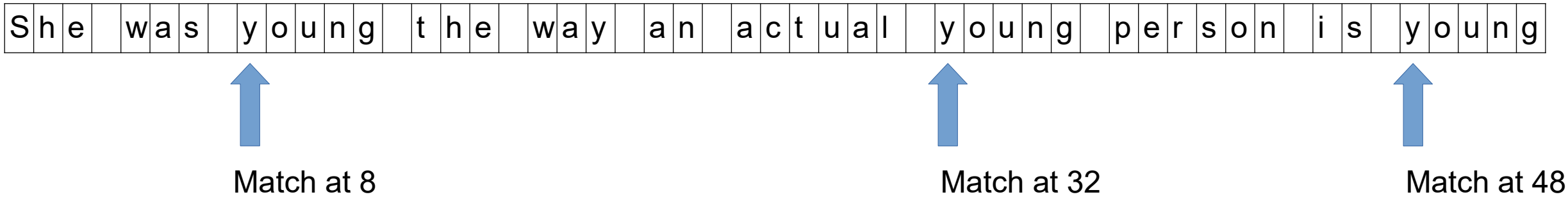
Can be case sensitive, or case insensitive. This lab will be case sensitive.

Instead of returning all the locations separately, we will return their sum.

Note that the first character in the string has index 0.

Example String Search

Consider searching for the string “young” in the following string:



We have matches at indices 8, 32, and 48. Since 8 + 32 + 48 = 88, the result of the search is 88.

CMPUT 229

Assignment

Overview of the lab

Overview:

In this lab, you must implement a single function to perform the described string search operation.

No other functions are needed.

stringSearch

Description:

This function computes the sum of all indices of the query string in the search string.

Parameters:

a0: Pointer to the query string.

a1: Pointer to the search string.

Return Value:

a0: Sum of the indices

CMPUT 229

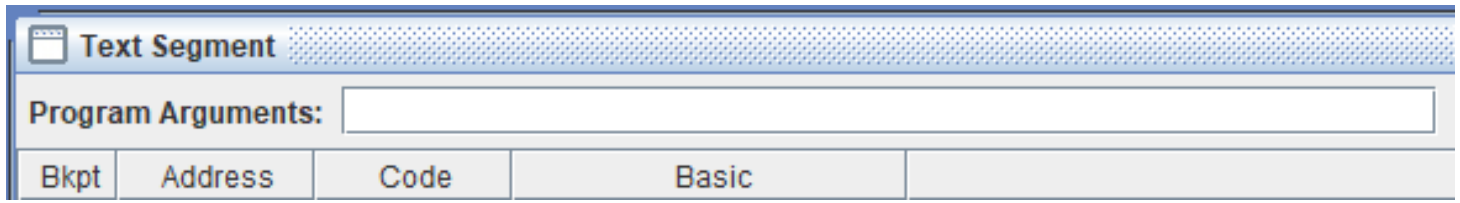
Testing

Testing your Solution

Give file path as program arguments:

Make sure the test file has the correct format (Next slide).

RARS may need the full path to the test file.



Format of the Test File

A test file contains an input to stringSearch:

The file must have two lines, both of which are ASCII strings.

The first line is the query string.

The second line is the search string.