Value	Letter	Value	Letter	Value	Letter	Value	Letter
65	A	78	N	97	a	110	n
66	B	79	O	98	b	111	o
67	C	80	P	99	c	112	p
68	D	81	Q	100	d	113	q
69	E	82	R	101	e	114	r
70	F	83	S	102	f	115	s
71	G	84	T	103	g	116	t
72	H	85	U	104	h	117	u
73	I	86	V	105	i	118	v
74	J	87	W	106	j	119	w
75	K	88	X	107	k	120	x
76	L	89	Y	108	l	121	y
77	M	90	Z	109	m	122	z
The values are decimal.							

1 Stack Memory and Pointer

Consider the following C program.

```
#include <stdio.h>
#include <stdlib.h>

// assume the size of the data types
// int: 4 bytes
// char: 1 byte
// double: 8 bytes
// pointer: 8 bytes

int f1(int a, char b)
{
  int t = a + b;
  // hint: you may need to use the ASCII table
  return t;
}

void f2(int * t)
{
  * t = 5;
  // empty line
}

int main(int argc, char ** argv)
{
  int a = 264;
  int b = ’A’;
  int c = 2019;
  c = f1(a, b);
  c = -111;
  // assign a new number so that the answer for f2 is not
  // affected by f1
  f2(& c);
  return EXIT_SUCCESS;
}
```

Please fill the stack as the program runs.
DNC means “do not care”. You do not need to answer.
Each address has A as the prefix.

1.1 `f1`

Please fill the stack memory when the program has just finished line 12, before executing line 14.

Frame	**Symbol**	**Address**	**Value**
`f1`	`t`	`A208`	**Question A**
	`b`	`A204`	*DNC*
	`a`	`A200`	264
	value address:		**Question B**
	return location:		`DNC`
`main`	`c`	`A120`	2019
	`b`	`A116`	'A'
	`a`	`A112`	264
	`argv`	`A104`	*DNC*
	`argc`	`A100`	*DNC*

1.2 `f2`

Please fill the stack memory when the program has just finished line 19, before function `f2` finishes.

Frame	**Symbol**	**Address**	**Value**
`f2`	`t`	`A208`	**Question C**
	return location:		line 33
`main`	`c`	`A120`	**Question D**
	`b`	`A116`	*DNC*
	`a`	`A112`	*DNC*
	`argv`	`A104`	*DNC*
	`argc`	`A100`	*DNC*