

## CLASS-13

### FUNCTIONS :

#### G. COALESCE

```
data a;  
input m1 m2 hl ol;  
contact = coalesce (m1, m2, hl, ol, 9999);  
cards;
```

1	4	6	7	} → customer 1 → customer 2 ⋮ customer-5	m1 = primary number m2 = secondary no. hl = home landline ol = office landline
3	.	2	1		
.	.	9	5		
.	.	.	.		
.	5	8	3		

;

run;

Suppose we want to contact the customers and for that we should know the phone numbers. Here the priority is set that first we should contact on their primary number i.e. - m1. If that is not available then contact on alternate number say m2 and so on with hl and ol. If we look at the data, suppose we contact

customer 1 on '1', <sup>null</sup>cust 2 on contact no. (m1) i.e 3, customer 3 on '9' as m1 and m2 are not given for <sup>null</sup>cust 3 and for cust 4 no contact no. is given.

Suppose we put dummy no. '9999' in customer 4 contacts. This no. '9999' will show that the contact no. of customer 4 is not updated in the database. and for 5<sup>th</sup> customer contact on 5. If we want to get this data and use 'if else if' then it will become a long program.

In this case we will use "coalesce" function. This function is used for computing values. It works on the missing values. It goes to the second argument if the first argument has null values. Eg: if we look at customer 3 data, m1 and m2 values are null so, it will automatically go to 'hl' i.e- 9 and read that data.

	m1	m2	hl	ol	contacts
1	1	4	6	7	1
2	3		2	1	3
3			9	5	9
4					9999
5		5	8	3	5

Output

Again if we look at the below data {another example}

data a;

set sasuser.admit;

x = coalesce (age, height, weight);

run;

Suppose if age is missing in a row <sup>x will take the value of</sup> ~~not~~ height,  
and if height is also missing, but weight.  
and if there will be some value in age so  
x will take the value of age.

So, basically It will only go to the second argument if the first argument has null values.



ID (cust ID)	MA (mailing address)	PA (Permanent addr)	OA (office addr)	TA (Temporary addr)	contact
1	Sec 46	□	□	□	
2	□	□	Sec 21	□	
3	□	□	□	□	NA

## 7. COALESCEC

→ character

If we want to send a letter to customer's mailing address. for customer 1 we will send to sector 46, for customer 2 will send to sector 21 and for customer 3, no address is given so we want the contact to take the values as 'NA'. for this we will use the function coalescec.

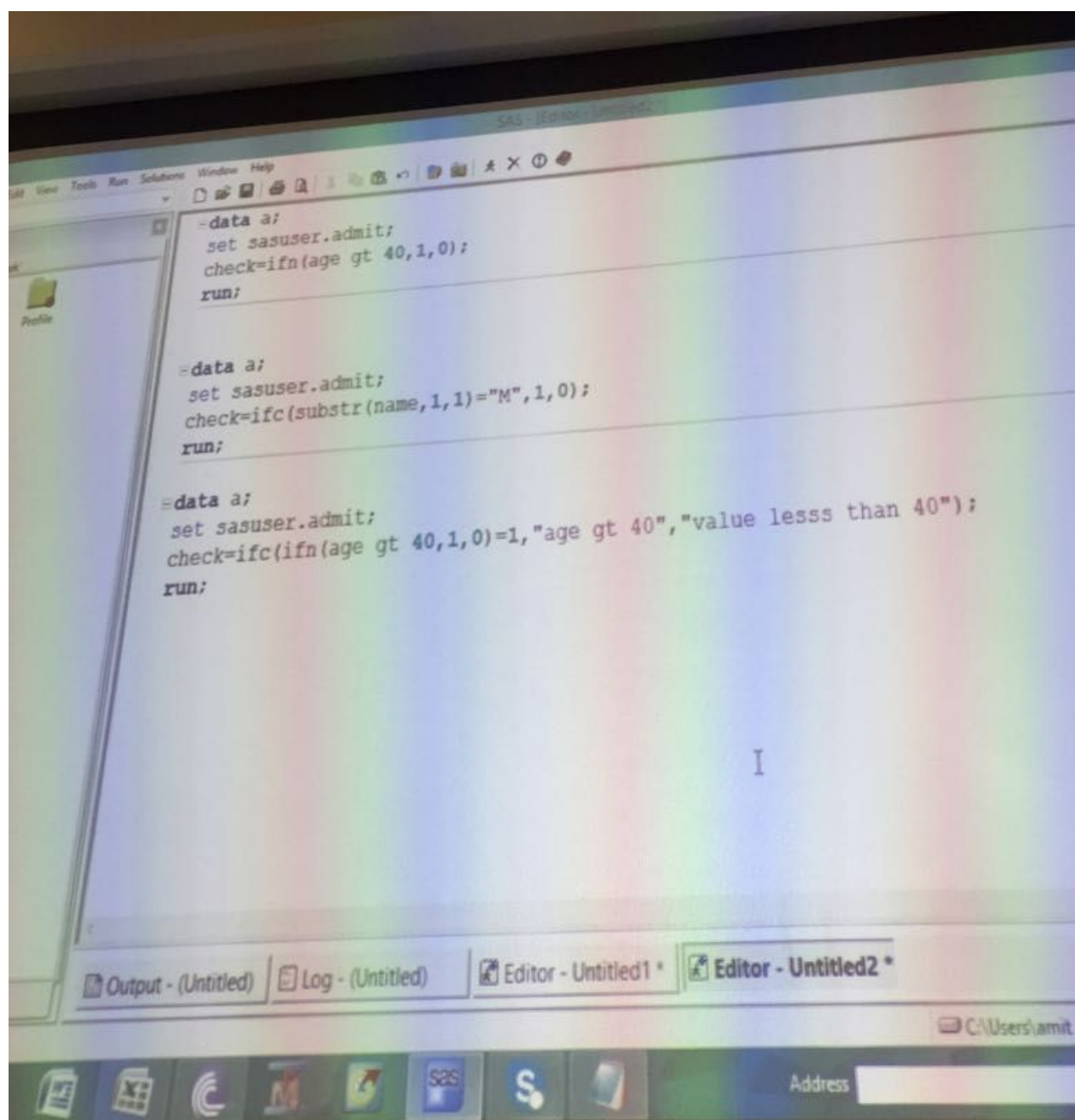
**NOTE**

If we want to replace anything in numeric use coalesce  
 & if we want to replace in character use coalescec.

```
data a;
input ma pa oa ta;
contact = coalescec (ma, pa, oa, ta, 'NA');
cards;
$refer above data;
run;
```

Output

	ma	pa	oa	ta	contact
1	Sec 46				Sec 46
2			Sec 21		Sec 21
3					'NA'



Hence they are ~~formal~~ <sup>other</sup>  
true/false  
False

⑧ Ifn → numeric

```
data a;  
set sasuser.admit;  
check = ifn (age gt 40, 1, 0);  
run;  
new  
(variable)
```

value true  
value false  
logical test

Ifn → Here n stands for number or numeric.  
It works as 'if else if'.

Here, we have created a new variable check with 'ifn' function and applied logical test 'age gt 40' and if the condition is true it will display the result as '1' else '0'.



Name	Sex	Age	Date	Height	Weight	ActLevel	Fee	check
Murray, W	M	27	1	72	168	HIGH	85.20	0
Almers, C	F	34	3	66	152	HIGH	124.80	0
Bonaventure, T	F	31	17	61	123	LOW	149.75	0
Johnson, R	F	43	31	63	137	MOD	149.75	1
LaMance, K	M	51	4	71	158	LOW	124.80	1
Jones, M	M	29	6	76	193	HIGH	124.80	0
Reberson, P	F	32	9	67	151	MOD	149.75	0
King, E	M	35	13	70	173	MOD	149.75	0
Pitts, D	M	34	22	73	154	LOW	124.80	0
Eberhardt, S	F	49	27	64	172	LOW	124.80	1
Nunnally, A	F	44	19	66	140	HIGH	149.75	1
Oberon, M	F	28	17	62	118	LOW	85.20	0
Peterson, V	M	30	6	69	147	LOW	149.75	0
Quigley, M	F	40	8	69	163	HIGH	124.80	0
Cameron, L	M	47	5	72	173	NA	124.80	1
Underwood, K	M	60	22	71	191	LOW	149.75	1
Takahashi, Y	F	43	29	65	123	MODY	124.80	1
Derber, B	M	25	23	75	188	HIGH	85.20	0
Ivan, H	F	22	20	63	139	LOW	85.20	0
Wilcox, E	F	41	16	67	141	HIGH	149.75	1
Warren, C	M	54	7	71	183	MOD	149.75	1

⑨ ~~Ifc~~ → character

data a;

set sasuser.admit;

check = ifc ( subste (name, 1, 1) = "M", 1, 0);

run;

Ifc → Here c stands for character.



Here we want that if the name starts with 'M' then display '1' else display '0'.

	Name	Sex	Age	Date	check
1	Seeta	F	27	1	0
2	Monu	M	25	3	1
3	Radha	F	24	17	0
4	Manisha	F	26	63	1
5	Minni	F	28	71	1

Output

Another example

```
data a;
set sasuser.admit;
check = ifc(ifn(age gt 40, 1, 0) = 1, "age gt 40", "value
less than 40");
run;
```

Here, we have used nested if functions. If we are using ifn then the conditions & arguments should also be numbers.

Ifc can take numeric as well as character arguments.

But if in case of 1, 0 we want 'yes' or 'no' then we cannot use ifn as it does not take character value.

ID	Name	Sex	Age	Date	Height	Weight	ActLevel	Fee	check
2458	Murray, W	M	27	1	72	168	HIGH	85.20	value less than 40
2462	Almers, C	F	34	3	66	152	HIGH	124.80	value less than 40
2501	Bonaventure, T	F	31	17	61	123	LOW	149.75	value less than 40
2523	Johnson, R	F	43	31	63	137	MOD	149.75	age gt 40
2539	LaMance, K	M	51	4	71	158	LOW	124.80	age gt 40
2544	Jones, M	M	29	6	76	193	HIGH	124.80	value less than 40
2552	Reberson, P	F	32	9	67	151	MOD	149.75	value less than 40
2555	King, E	M	35	13	70	173	MOD	149.75	value less than 40
2563	Pitts, D	M	34	22	73	154	LOW	124.80	value less than 40
2568	Eberhardt, S	F	49	27	64	172	LOW	124.80	age gt 40
2571	Nunnelly, A	F	44	19	66	140	HIGH	149.75	age gt 40
2572	Oberon, M	F	28	17	62	118	LOW	85.20	value less than 40
2574	Peterson, V	M	30	6	69	147	LOW	149.75	value less than 40
2575	Quigley, M	F	40	8	69	163	HIGH	124.80	value less than 40
2578	Cameron, L	M	47	5	72	173	NA	124.80	age gt 40
2579	Underwood, K	M	60	22	71	191	LOW	149.75	age gt 40
2584	Takahashi, Y	F	43	29	65	123	MODY	124.80	age gt 40
2586	Derber, B	M	25	23	75	188	HIGH	85.20	value less than 40
2588	Ivan, H	F	22	20	63	139	LOW	85.20	value less than 40
2589	Wilcox, E	F	41	16	67	141	HIGH	149.75	age gt 40
2595	Warren, C	M	54	7	71	183	MOD	149.75	age gt 40

Log - (Untitled)

Editor - Untitled1 \*

Explorer

VIEWTABLE: Work-A

## CAT function

Cat function concatenates the value of variables.

data a;

set sasuser. admit;

x = cat (name, sex);

run;

	x	
1	Rohan	M
2	Preety	F
3	Sadhna	F

Output

data a;

x1 = 'a';

x2 = 'b';

x3 = 'c';

string = cat ( of x1 - x3 );

run;

	x1	x2	x3	string
1	a	b	c	abc



### (10) LOWCASE

It converts the string into small letters or lower case.

```
data a;  
x = 'AMIT';  
y = lowercase(x);  
run;
```

	x	y
1	AMIT	amit

### (11) UPCASE

It converts the string into capital letters or upper case.

```
data a;  
x = 'amit';  
y = upcase(x);  
z = upcase('Ram is a boy');  
run;
```

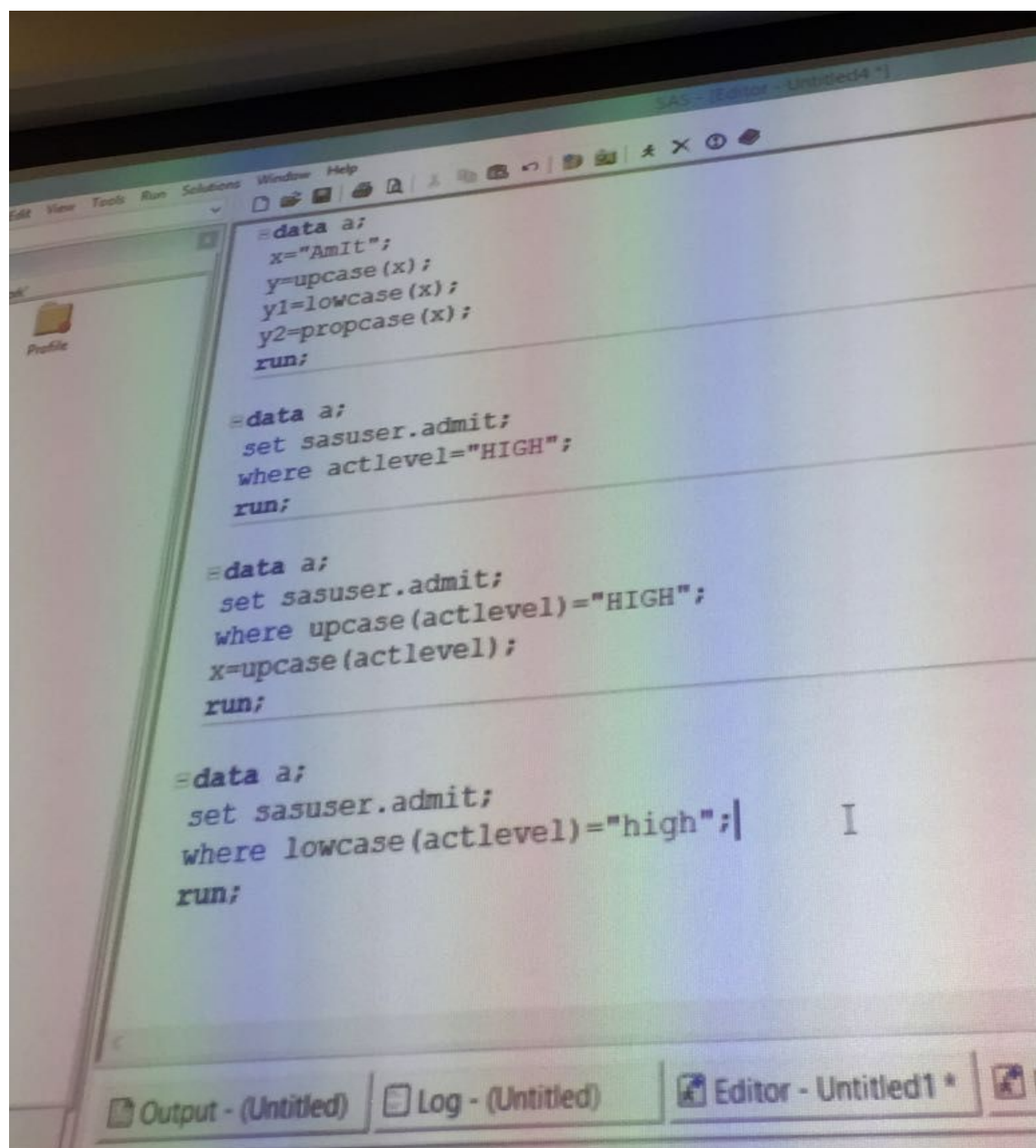
	x	y	z
1.	amit	AMIT	RAM IS A BOY

## 12) PROPCASE

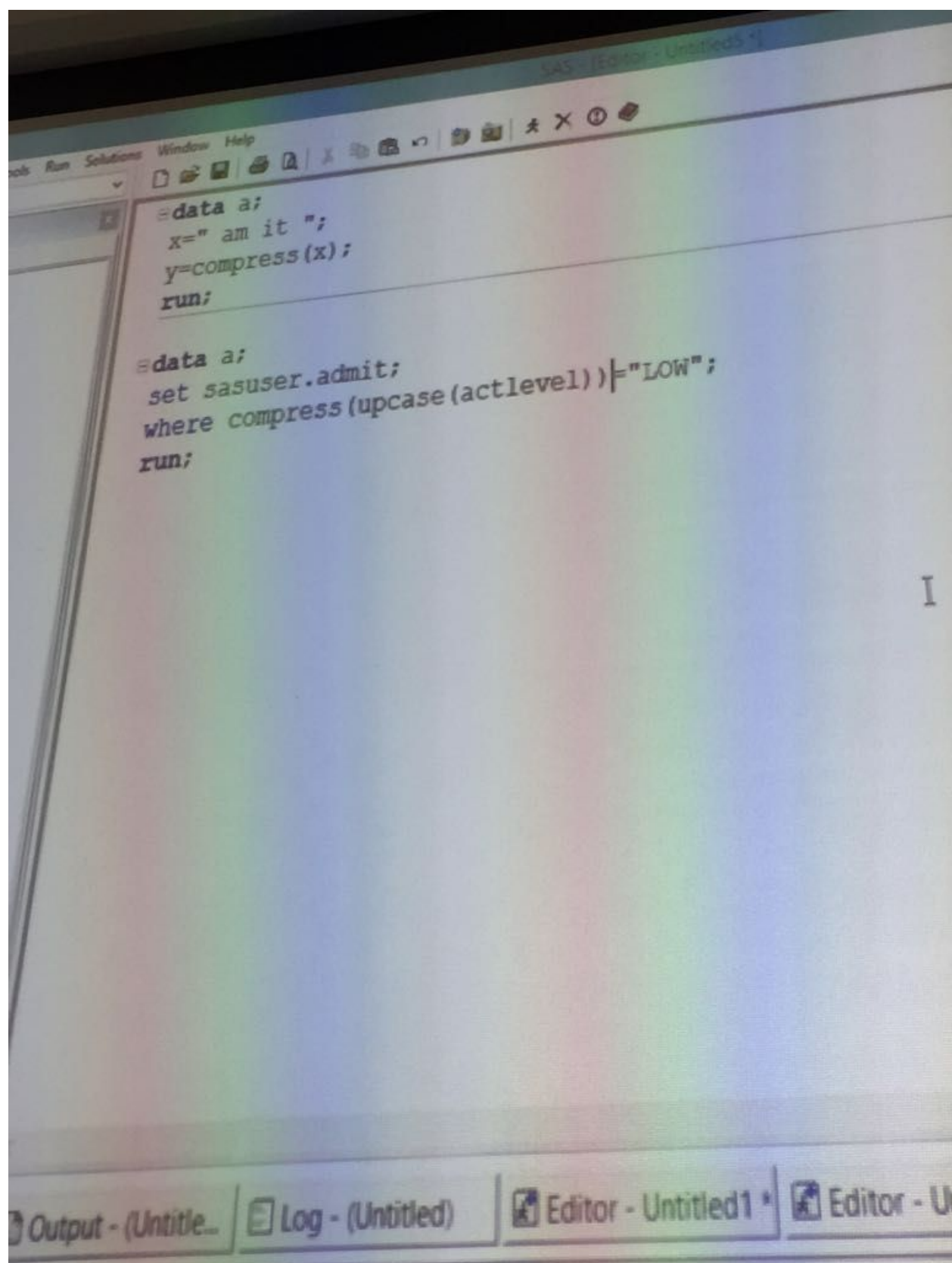
It converts the string into proper case. First letter of each word is in uppercase and all other letters in lower case.

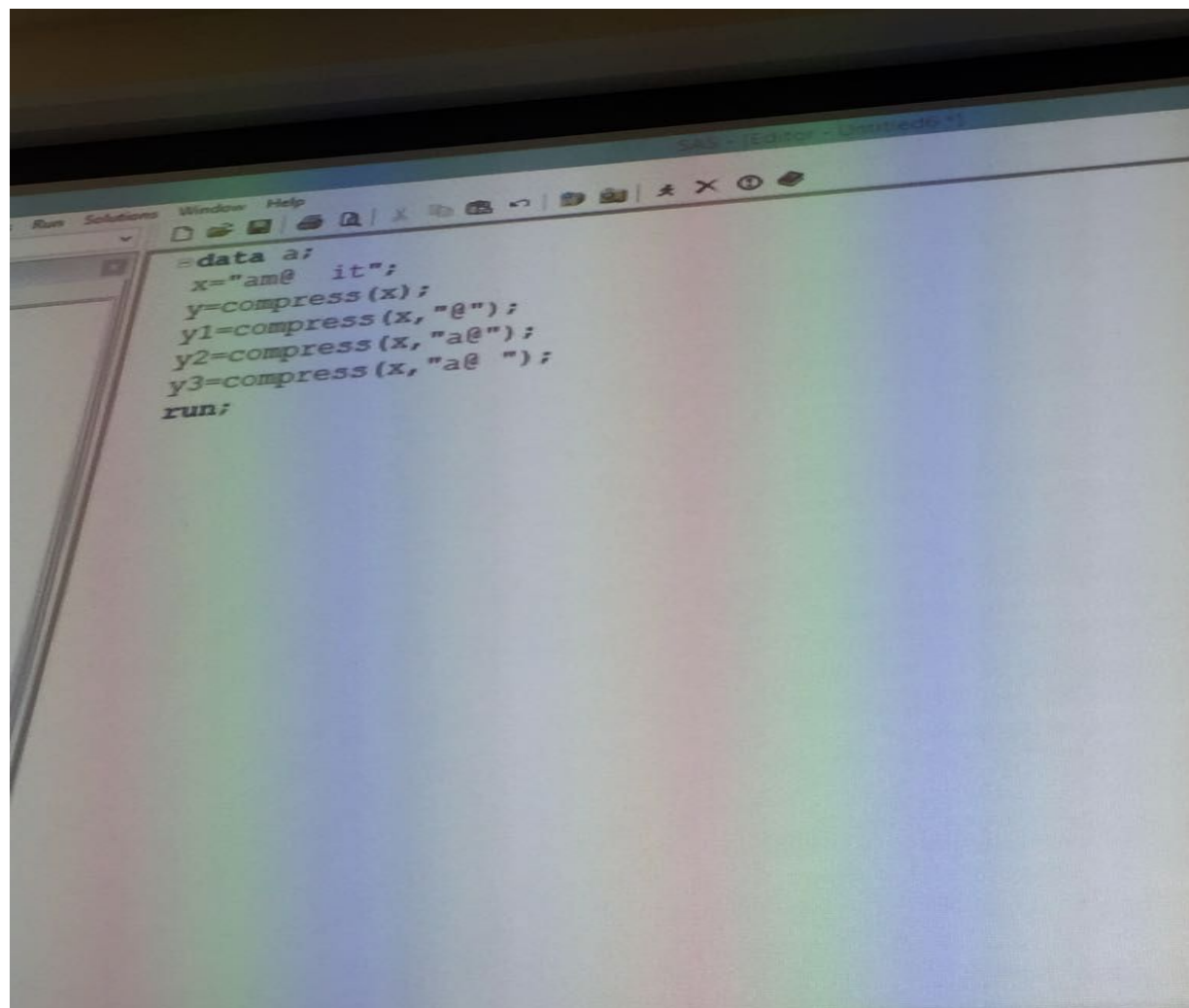
```
data a;  
x = 'AMIT';  
y = propcase(x);  
z = propcase('apple is red');  
run;
```

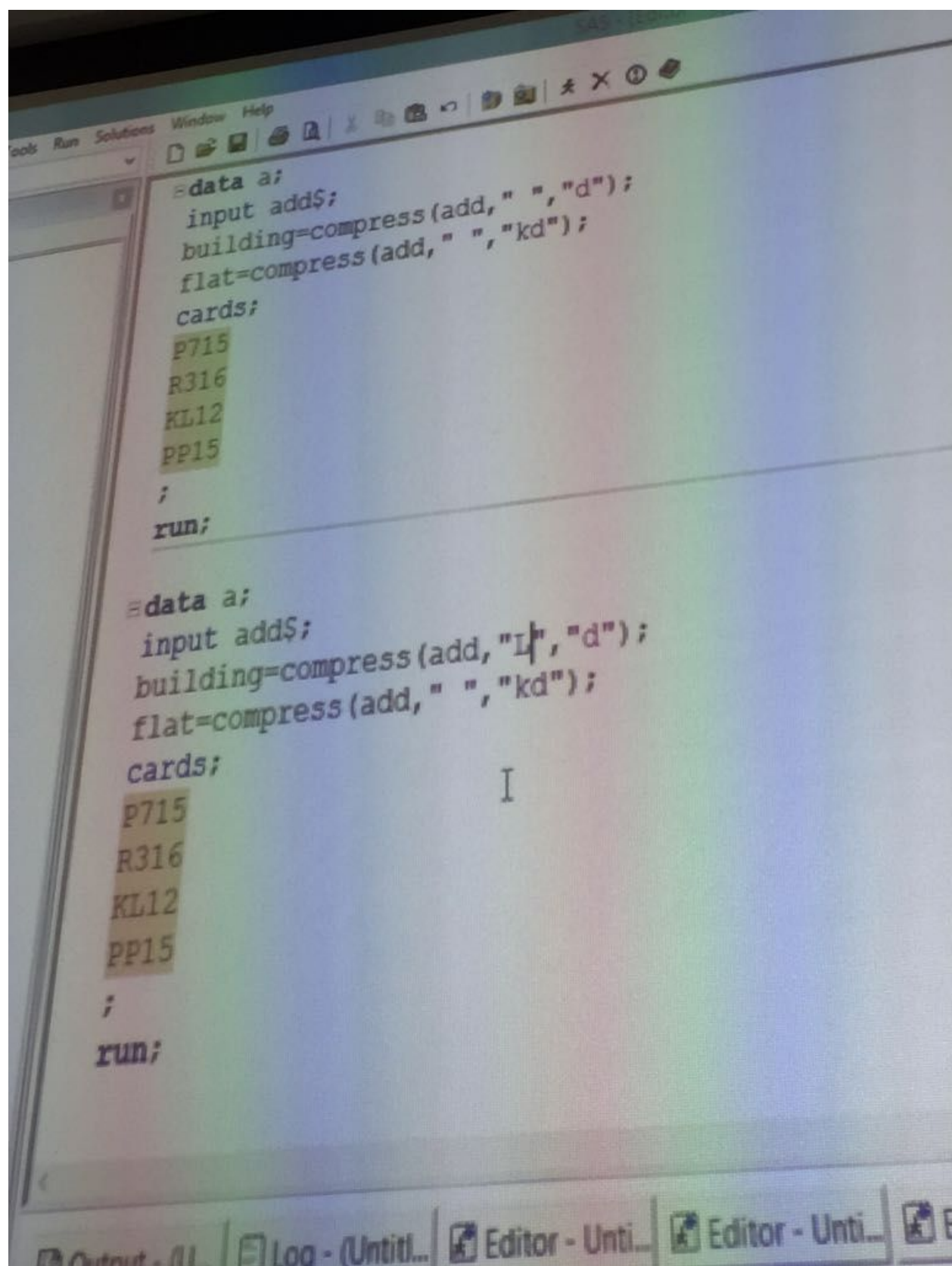
	x	y	z
1	AMIT	<u>A</u> mit	<u>A</u> pple <u>I</u> s <u>R</u> ed













COMPRESS (optimize space)

$x = \text{" -- am -- it -- "};$

leading space

↓  
Intermediate  
space

→ Trailing space

$$y = \underline{\text{compress}}(x) \rightarrow (\text{emit}) \quad (\text{output})$$

(It removes leading, trailing and intermediate all the blanks)

$y = \underline{\underline{\text{strip}}}(x) \rightarrow \text{am} - \text{it} \text{ (output)}$

(It removes leading and trailing blanks)

$y = \underline{\text{trim}}(x) \rightarrow \text{---ám---it (output)}$

(It removes only trailing blanks)

### Example ①

data a;

x = " am @ it ";

y = compress(x); → Remove white space.

y1 = compress(x, "@"); → (explicit argument @ is given so will remove @)

y2 = compress(x, "a@");

y3 = compress(x, "a@ ");

run;

Output

	x	y	y1	y2	y3
1	am @it	am@it	am it	m it	mit

### Another example

data a;

set sasuser.admit;

where compress(upcase(factlevel)) = "Low";

run;

note: whenever we do string matching, minimum function we should always use is 'compress' and 'upcase'.

Note { from example ① } (Interview Question)

compress takes both 1 and 2 arguments. If it takes 1 argument it will delete white spaces and if it takes 2 arguments then it explicitly runs after 2nd argument byte by byte or character by character.

Another example

```
data a;  
input add $;  
building = compress (add, " ", "d");  
flat = compress (add, " ", "kd");  
cards;  
P 715  
R 316  
KL 12  
PP 15  
;  
run;
```

delete digits  
keep digits

P-block/building ~~store~~ House no 715.  
These are the addresses.

Suppose we want to make two variables -

- ① building - in which the values P, R, KL, PP should go
- ② flat - in which the values 715, 316, ... should go.

By using 'd' → delete digits in variable 'building'  
 and using 'kd' → keep digits in variable 'flat'.  
 we can separate out the alpha and numeric  
 values.

Output

	add	building	flat
1	P715	P	715
2	R316	R	316
3	KL12	KL	12
4	PP15	PP	15

Output

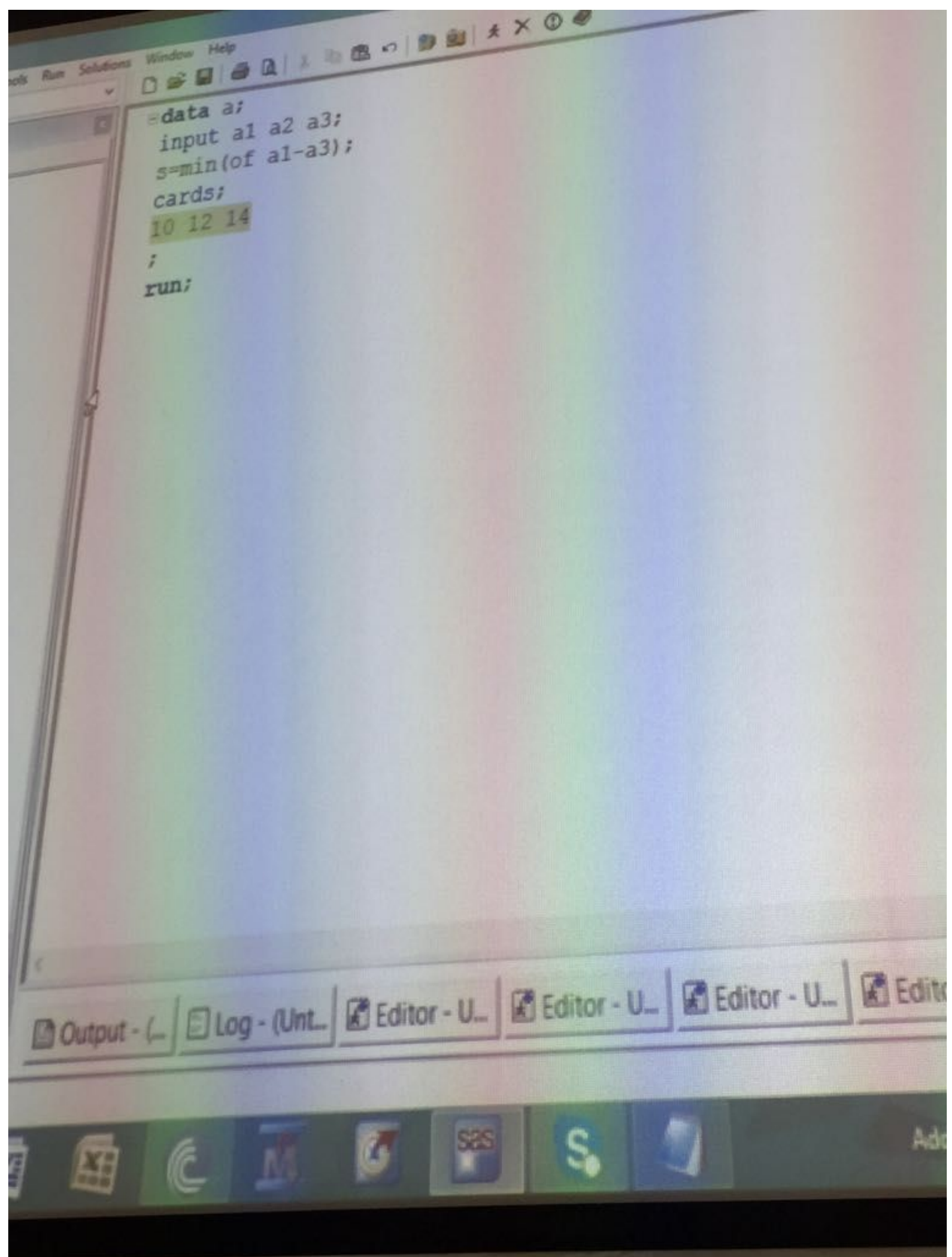
```
data a;
input add $;
building = compress (add, "L", "d");
flat = compress (add, " ", "kd");
cards;
P 715
R 316
KL 12
PP 15
;
run;
```

→ If we put 2<sup>nd</sup> argument  
 as 'L' in variable building  
 then it will delete L from  
 the value 'KL' and will display  
 only 'K'.



## Output

	add	building	flat
1	P715	P	715
2	R316	R	316
3	KL12	K	12
4	PP15	PP	15

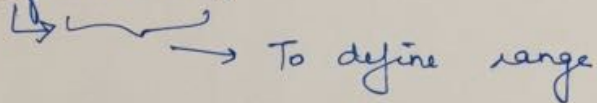


Mean function { calculate the mean of values }

data a;

input a1 a2 a3;

\*S = mean(of a1 - a3)

↳  To define range

cards;

10 12 14

;

run;

	a1	a2	a3	S
1	10	12	14	12

Output

Avg function

data a;

input a1 a2 a3;

avg = mean(of a1 - a3)

cards;

10 12 14

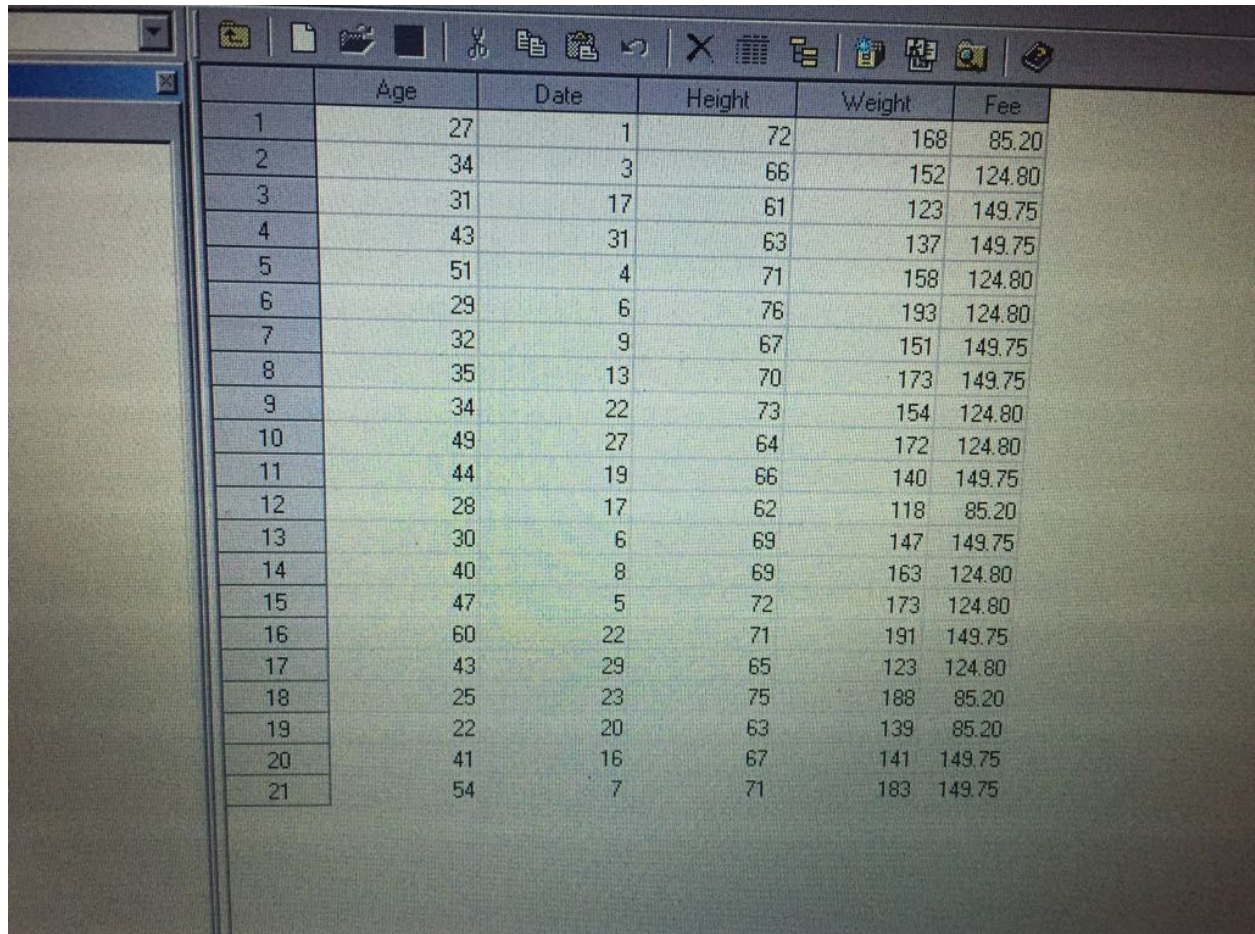
;

run;

	a1	a2	a3	avg
1	10	12	14	12

```
data a (keep = _numeric_);  
set sasuser.admit;  
run;
```

( It will shew all the numeric variables ).



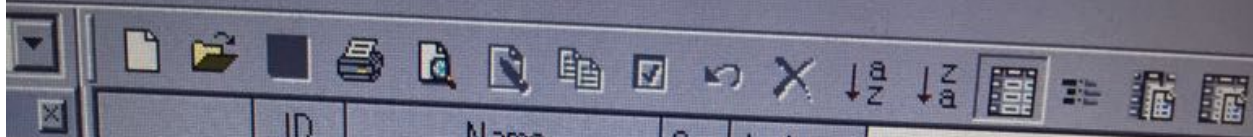
	Age	Date	Height	Weight	Fee
1	27	1	72	168	85.20
2	34	3	66	152	124.80
3	31	17	61	123	149.75
4	43	31	63	137	149.75
5	51	4	71	158	124.80
6	29	6	76	193	124.80
7	32	9	67	151	149.75
8	35	13	70	173	149.75
9	34	22	73	154	124.80
10	49	27	64	172	124.80
11	44	19	66	140	149.75
12	28	17	62	118	85.20
13	30	6	69	147	149.75
14	40	8	69	163	124.80
15	47	5	72	173	124.80
16	60	22	71	191	149.75
17	43	29	65	123	124.80
18	25	23	75	188	85.20
19	22	20	63	139	85.20
20	41	16	67	141	149.75
21	54	7	71	183	149.75



```
data a (keep = _character _);  
set sasuser.admit;  
run;
```

(It will show all the character variables);

Window Help



	ID	Name	Sex	ActLevel
1	2458	Murray, W	M	HIGH
2	2462	Almers, C	F	HIGH
3	2501	Bonaventure, T	F	LOW
4	2523	Johnson, R	F	MOD
5	2539	LaMance, K	M	LOW
6	2544	Jones, M	M	HIGH
7	2552	Reberson, P	F	MOD
8	2555	King, E	M	MOD
9	2563	Pitts, D	M	LOW
10	2568	Eberhardt, S	F	LOW
11	2571	Nunnelly, A	F	HIGH
12	2572	Oberon, M	F	LOW
13	2574	Peterson, V	M	LOW
14	2575	Quigley, M	F	HIGH
15	2578	Cameron, L	M	NA
16	2579	Underwood, K	M	LOW
17	2584	Takahashi, Y	F	MODY
18	2586	Derber, B	M	HIGH
19	2588	Ivan, H	F	LOW
20	2589	Wilcox, E	F	HIGH
21	2595	Warren, C	M	MOD

Output - (Untitled)

Log - (Untitled)

Editor - Untitled1 \*

e mode.

- [VIEWTABLE: Wor...

```
data a (keep = name -- weight);  
set sasuser.admit;  
run;
```

( It will show variables from name to weight ).

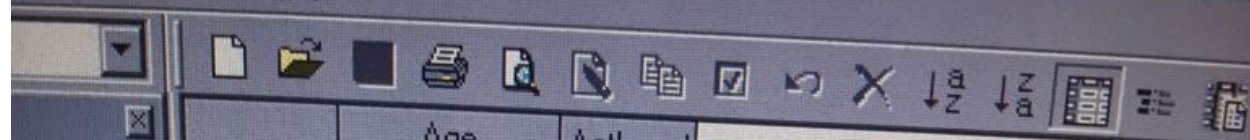


	Name	Sex	Age	Date	Height	Weight
1	Murray, W	M	27	1	72	168
2	Almers, C	F	34	3	66	152
3	Bonaventure, T	F	31	17	61	123
4	Johnson, R	F	43	31	63	137
5	LaMance, K	M	51	4	71	158
6	Jones, M	M	29	6	76	193
7	Reberson, P	F	32	9	67	151
8	King, E	M	35	13	70	173
9	Pitts, D	M	34	22	73	154
10	Eberhardt, S	F	49	27	64	172
11	Nunnally, A	F	44	19	66	140
12	Oberon, M	F	28	17	62	118
13	Peterson, V	M	30	6	69	147
14	Quigley, M	F	40	8	69	163
15	Cameron, L	M	47	5	72	173
16	Underwood, K	M	60	22	71	191
17	Takahashi, Y	F	43	29	65	123
18	Derber, B	M	25	23	75	188
19	Ivan, H	F	22	20	63	139
20	Wilcox, E	F	41	16	67	141
21	Warren, C	M	54	7	71	183

```
data a (keep = a:);
set sasuser.admit;
run;
```

(It will show variables starting from 'a');





	Age	ActLevel
1	27	HIGH
2	34	HIGH
3	31	LOW
4	43	MOD
5	51	LOW
6	29	HIGH
7	32	MOD
8	35	MOD
9	34	LOW
10	49	LOW
11	44	HIGH
12	28	LOW
13	30	LOW
14	40	HIGH
15	47	NA
16	60	LOW
17	43	MODY
18	25	HIGH
19	22	LOW
20	41	HIGH
21	54	MOD

Output - (Untitled)

Log - (Untitled)

Editor - Untitled1 \*

browse mode.

SAS - [VIEWTABLE: W...



```
data a (keep = a: w:);
```

```
set sasuser.admit;
```

```
run;
```

(It will show variables starting from 'a' as well as 'w').

Solutions Window Help			
	Age	Weight	ActLevel
1	27	168	HIGH
2	34	152	HIGH
3	31	123	LOW
4	43	137	MOD
5	51	158	LOW
6	29	193	HIGH
7	32	151	MOD
8	35	173	MOD
9	34	154	LOW
10	49	172	LOW
11	44	140	HIGH
12	28	118	LOW
13	30	147	LOW
14	40	163	HIGH
15	47	173	NA
16	60	191	LOW
17	43	123	MODY
18	25	188	HIGH
19	22	139	LOW
20	41	141	HIGH
21	54	183	MOD

Output - (Untitled) Log - (Untitled) Editor - Untitled

ed in browse mode.

SAS



