Tshwane University of Technology
We empower people

FORMAL SUMMATIVE ASSESSMENT INSTRUCTIONS:

1. All rules and regulations regarding student conduct and behaviour, as agreed upon by students at registration and as otherwise stated by the Tshwane University of Technology, apply.

2. A student card or proof of registration, along with an original identification document, must be presented to invigilators.

3. Ensure that your student number is correctly indicated on all submissions (paper-based or online) and that the attendance record is signed as required.

4. If a student is suspected of any form of cheating or plagiarism, either during the assessment or afterwards, the examiner, in their professional judgment and after consultation with the appointed moderator, may, at minimum, award a zero mark for the assessment. Further action may be taken against the student.

DEPARTMENT OF
ELECTRICAL ENGINEERING

FACULTY OF
ENGINEERING AND THE BUILT ENVIRONMENT

MODULE CODE
ES216AB

MODULE NAME
ENGINEERING SOFTWARE DESIGN A

ASSESSMENT NAME
Evaluation 2

EXAMINER:	Mr D. Engelbrecht Prof J.A. Jordaan	**MARKS:**	50 points
MODERATOR:	Mr A.J. Smith	**PAGES:**	11 (incl. cover)
DATE:	07 April 2025 09:30 to 11:30	**TIME:**	2 hours

STUDENT NUMBER:								

SURNAME:	**INITIALS:**	**SIGNATURE:**

NOTE THE FOLLOWING:

1. **Plagiarism Policy:**
   Original work is required. We will use similarity detection software to review all student submissions for plagiarism. Ensure your work is your own.
2. **Internet Protocol (IP) Tracking:**
   IP addresses will be recorded and checked to verify that you have uploaded your work from the correct TUT laboratory.
3. **No External Devices:**
   The use of USB or other external devices is prohibited during the evaluation.
4. **Internet Access:**
   External internet access is not permitted.
5. **Evaluation Content:**
   This evaluation will cover topics from Unit 1 to Unit 5.
6. **Programming Language:**
   Write your program in C, adhering to structured programming principles.
7. **Editing Requirements**:
   Your program must comply with all specified requirements. Refer to the appendices and attachments for more details.
8. **Submission Format:**
   Submit your source code header file in the following format:
   <student number>.h , for example, 217123456.h
   <u>ONLY YOUR STUDENT NUMBER!</u> Do not add other text.
9. **Submission Upload:**
   Use the dedicated upload link on MyTUTorD2L to upload your C header file code only. While multiple submissions are allowed, only the latest submission will be retained. If you upload the wrong file by mistake, simply re-upload the correct one, and the previous submission will be overwritten.
10. **Backup And Save:**
    Remember to save your work on the PC D: Drive and save regularly throughout the evaluation. In the event of PC malfunction or power failure, only 5 minutes (depending on the case) extra time will be allotted.

QUESTION:

To complete the functionality of the "**Random Number Generator**" C programming application, several functions are needed.

The "**Random Number Generator**" application requires the generation of 20 random values (between -10 and +10), which are stored in the "**Arr**" array. Subsequently, each function is explained below and how it should work. Your application must be in a **continuous** loop and us a **SWITCH CASE selection structure**

The application starts by calling the "Menu" function that displays the heading and menu options for the user to select from.

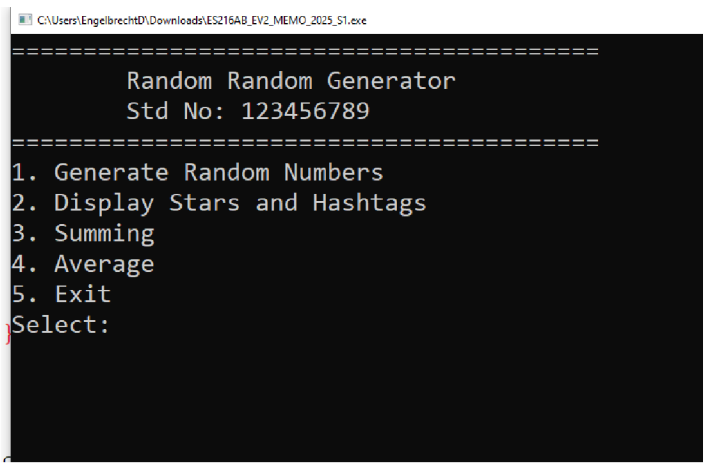
```
=============================================
                    Random Number Generator
                    Std No: 123456789
=============================================
```

1. Generate Random Numbers
2. Display Stars and Hashtags
3. Summing
4. Average
5. Exit

1. Menu Function

char Menu(void)

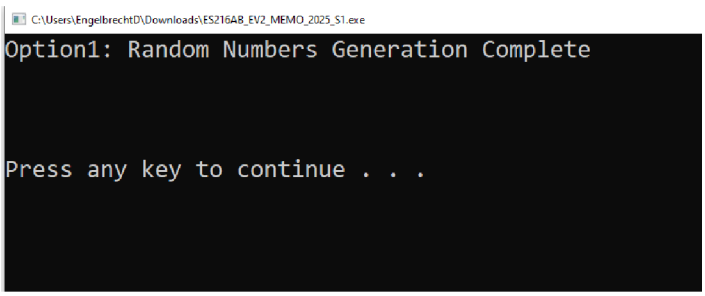
The "**Menu**" function creates the heading and menu with all the options. It returns the char value of the user's selection.



2. GenerateRandomNumbers Function

void GenerateRandomData (int Arr[], int Size);

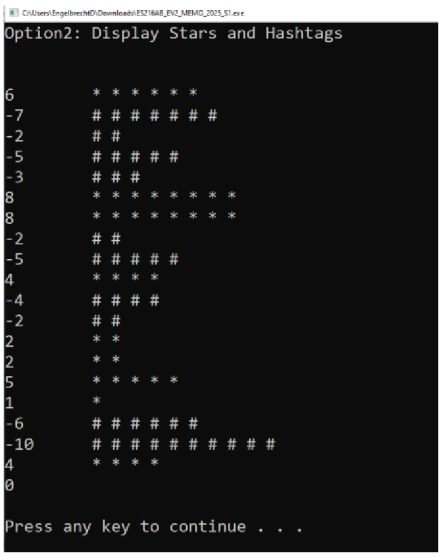
The "**GenerateRandomNumbers**" function initialises an array with random numbers. This function takes two parameters: an integer array "**Arr[]**" and the size of this array "**Size**". It uses the "**srand**" function to set the seed of the "**rand**" function based on the current time (ensuring different random values on each program execution). It then populates the array with random integers between -10 and +10, by iterating from 0 up to "**Size-1**" and assigning each array element a random number.
(No printf() or scanf() statements may be used in the **GenerateRandomNumber** function)



3. DisplayStarsandHashtags Function

void DisplayStarsandHashtags(int Arr[], int Size);

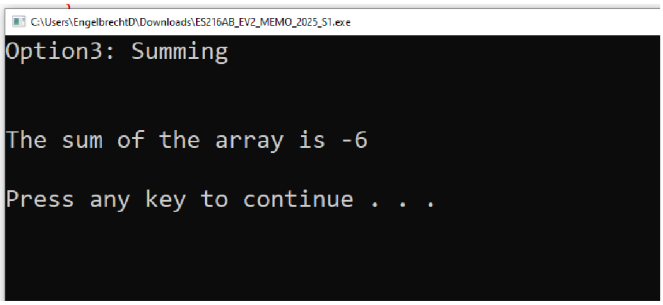
The "**DisplayStarsandHashtags**" function takes two parameters: an integer array "**Arr[]**" and the size of this array "**Size**".to print the value of each element in the array. The function then prints the corresponding amount of stars for positive values and the corresponding amount of hashtags for negative values next to the bumber. No symbols are printed for zero values.



4. Summing Function

int Summing (int Arr[], int Size);

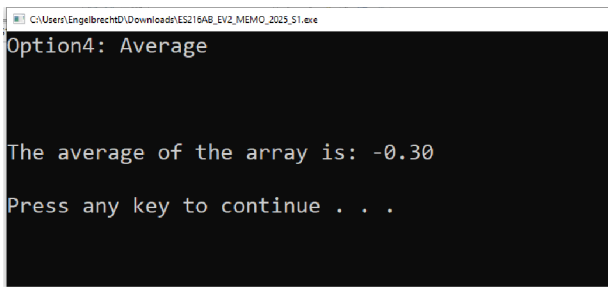
The "**Summing**" function takes two parameters: an integer array "**Arr[]**" and the size of this array "**Size**". The function calculates the sum of the entire array and returns the value.
(No printf() or scanf() statements may be used in the **Average** function)



5. Average Function

float Average (int Sum);

The "**Average**" function takes one parameters: an integer "**Sum**"". The function calculates the average using the "Sum" input value and returns the average value.
(No printf() or scanf() statements may be used in the **Average** function)



Option '5' on the menu should close the program, Check for invalid selections and use system("cls") and system("pause") to make your program user friendly.

```
C:\Users\EngelbrechtD\Downloads\ES216AB_EV2_MEMO_2025_S1.exe
========================================
        Random Random Generator
        Std No: 123456789
========================================
1. Generate Random Numbers
2. Display Stars and Hashtags
3. Summing
4. Average
5. Exit
Select: 8
Invalid choice........

Press any key to continue . . .
```

```
C:\Users\EngelbrechtD\Downloads\ES216AB_EV2_MEMO_2025_S1.exe
========================================
        Random Random Generator
        Std No: 123456789
========================================
1. Generate Random Numbers
2. Display Stars and Hashtags
3. Summing
4. Average
5. Exit
Select: 5


Program ends........


Press any key to continue . . .
```

Instructions for Adherence: Proper adherence to provided instructions, control structures, naming conventions, and structured programming principles is crucial for obtaining full marks. Even if the program functions correctly, it does not guarantee full marks. Proper indentation and comments are essential for clarity and understanding. No global variables may be implemented in the final solution.

IMPLEMENT THE FOLLOWING FLOW FORMAT:

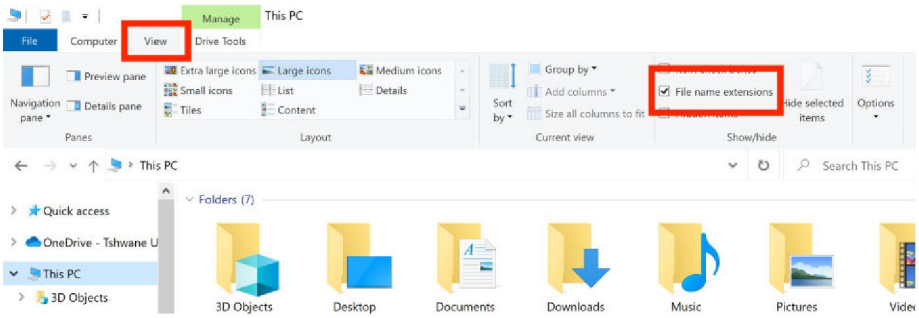
0. Libraries
1. Continuous Loop
2. Menu Function
3. Switch Case Selection Structure
4. GenerateRandomNumbers Function
5. DisplayStarsandHashtags Function
6. Summing Function
7. Average Function

Use the following function prototypes:

```
char Menu (void);
void GenerateRandomNumbers (int Arr[], int Size);
void DisplayStarsandHashtags (int Arr[], int Size);
int Summing (int Arr[], int Size);
float Average (int Sum);
```

ANNEXURE A – HOW TO RUN THE SHOWCASE

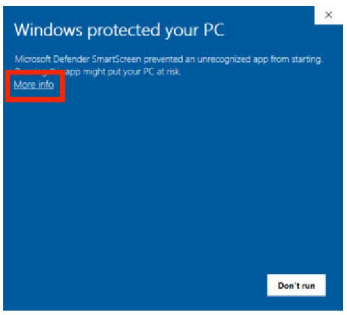
1. Enable “File name extensions” (see highlighted in red)



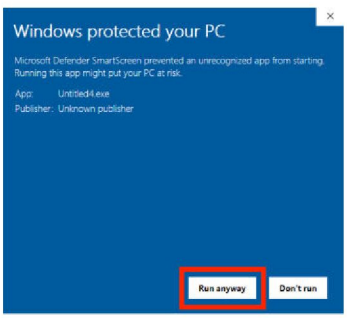
2. Change the file extension from **“Example.old”** to **“Example.exe”**

3. Run the **“Example.exe”** by double-clicking on the icon.

4. Windows may show the following. Click on **“More info”**



5. Click on **“Run anyway”**



ANNEXURE B – MARK ALLOCATION

Note: Score range is 0 - 4 which is: 0-none, 1-poor, 2-average, 3-good, 4-excellent

TEST RUBRIC	SCORE [0-4]	WEIGHT [%]
C CODE & OUTPUT EVALUATION		**50**
1. Menu Function		**6**
2. GenerateRandomNumbers Function		**8**
3. DisplayStarsandHashtags Function		**8**
4. Summing Function		**6**
5. Average Function		**4**
Switch Case and While loop correct		**4**
Invalid Selection		**2**
Exit Programs		**2**
User Friendliness		**5**
No Compile or Runtime errors		**5**
TOTAL		**50**

Graduate Attribute	GA Number	GA Score [0-5]
Engineering Professionalism	GA10	
Application of scientific and engineering knowledge	GA2	
Engineering methods, skills, tools, including information technology	GA5	
Impact of Engineering Activity	GA7	

ANNEXURE D – INFORMATION SHEET

Libraries:	<stdio.h> , <stdlib.h> , <time.h> , <math.h>
Data types:	void, char, short, int, float, double
Data Type modifiers:	const, auto, static, unsigned, signed
Arithmetic operators:	* / % + -
Relational operators:	< <= > >= == !=
Assignment operator:	= += -= *= /= %= &= ^= |= <<= >>=
Logic operators:	&& || !
Bitwise logic operators:	& | ^ ~ << >>
Pointer operators:	Derefernce: * Address: &
Control Structures:	
IF Selection:	if (condition) { … };
IF ELSE Selection:	if (condition) { … } else { … };
SWITCH Selection:	switch (control variable) { case 'value': … ; break; default: … ; break; }
FOR Loop:	for (initial value of control variable; loop condition; increment of control variable) { … }
WHILE Loop:	while (condition) { … };
DO WHILE loop:	do { … } while (condition);
Functions:	return_data_type function_name (parameters) { … };
Common Library Functions:	printf() , scanf() , rand() , srand() , time() , isalpha() , isdigit() , getchar() , getch(), strcpy()
Arrays:	One dimensional:data_type variable_name[size]; Two dimensional: data_type variable_name [x_size][y_size];

ANNEXURE E – ASCII TABLE

Dec	Hx	Oct	Char		Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL	(null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH	(start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX	(start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX	(end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT	(end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ	(enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK	(acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL	(bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS	(backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB	(horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF	(NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT	(vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF	(NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR	(carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO	(shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI	(shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE	(data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1	(device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2	(device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3	(device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4	(device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK	(negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN	(synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB	(end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN	(cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM	(end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB	(substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC	(escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS	(file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS	(group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS	(record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US	(unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL