TUTORIAL 8

Tutorial Task(s)

1. Why pointer is used?

Pointers are used to store and manage the addresses of dynamically allocated blocks of memory. Such blocks are used to store data objects or arrays of objects.

2. Why pointer should have data types when their size is always 4 bytes(in a32 bit machine), irrespective of the variable they are pointing to?

An unsigned integer is allocated 4 bytes of memory for its storage on a typical 32-bit system. Thus, pointers to all types of data occupy the same size of memory because the value of a pointer is the memory address – an unsigned integer. The reason why you need the data type for pointers is because the compiler has to know what the size of the memory cell is, among others, the pointer is pointing to. Also type safety cannot be ensured w/o the type. Also, you would have to typecast every pointer when accessing structures from the pointer.

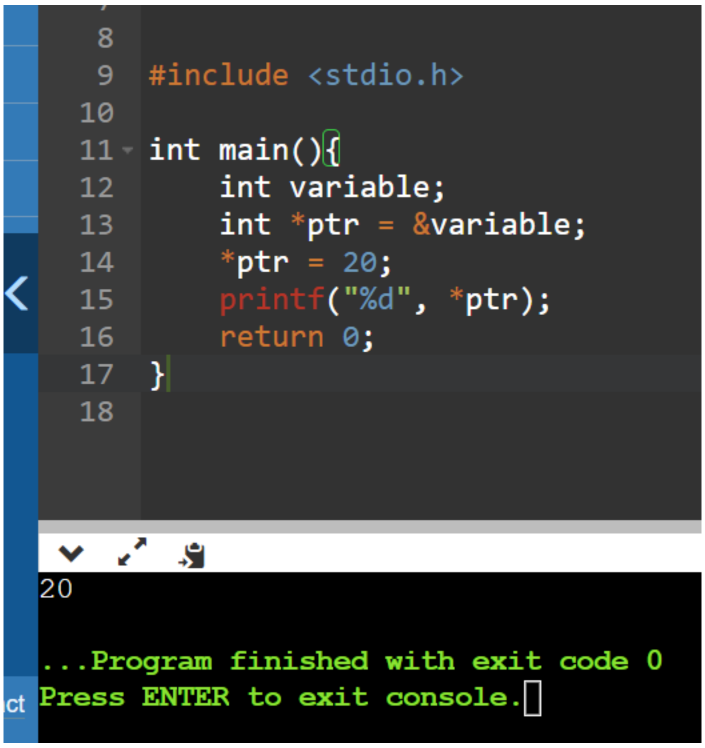
3. What is wrong with following code segment?

a. Int *p;

b. *p=10;

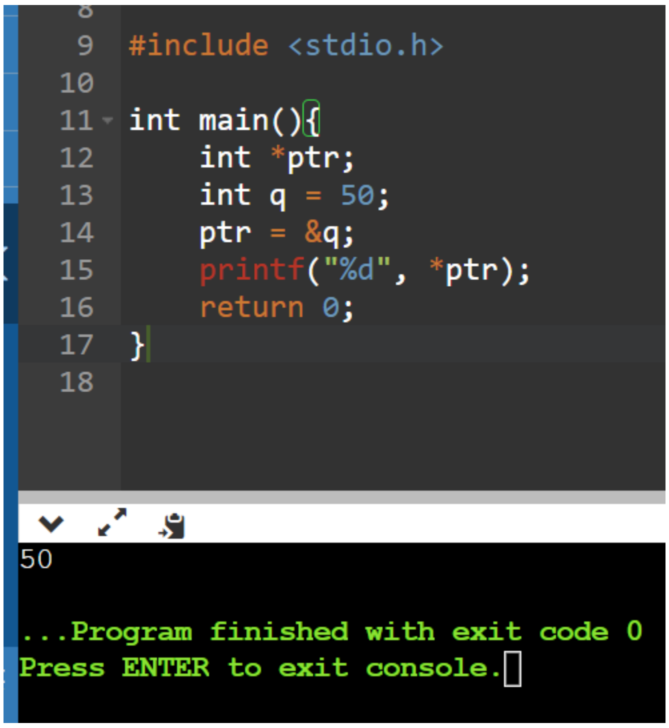
*p=10 is wrong way of assigning a value to a pointer. It should be done like this-

Samuela Abigail
71762108039



```c
#include <stdio.h>

int main(){
    int variable;
    int *ptr = &variable;
    *ptr = 20;
    printf("%d", *ptr);
    return 0;
}
```

```
20

...Program finished with exit code 0
Press ENTER to exit console.
```



```c
#include <stdio.h>

int main(){
    int *ptr;
    int q = 50;
    ptr = &q;
    printf("%d", *ptr);
    return 0;
}
```

```
50

...Program finished with exit code 0
Press ENTER to exit console.
```

Samuela Abigail
71762108039

4. Does C have "pass by reference" feature?

C doesn't have pass by reference, it can only be emulated using pointers.

5. What is wild pointer in C?

Pointers store the memory addresses. Wild pointers are different from pointers i.e. they also store the memory addresses but point the unallocated memory or data value which has been deallocated. Such pointers are known as wild pointers. A pointer behaves like a wild pointer when it is declared but not initialized.

6. Is a null pointer same as an uninitialized pointer?

A null pointer should not be confused with an uninitialized pointer: a null pointer is guaranteed to compare unequal to any pointer that points to a valid object. However, depending on the language and implementation, an uninitialized pointer may not have any such guarantee.

7. What are the uses of null pointers?

Some uses of the null pointer are: a) To initialize a pointer variable when that pointer variable isn't assigned any valid memory address yet. b) To pass a null pointer to a function argument when we don't want to pass any valid memory address. c) To check for null pointer before accessing any pointer variable

8. Is NULL always defined as 0?

The null pointer constant is always 0. The NULL macro may be defined by the implementation as a naked 0 , or a cast expression like (void *) 0 , or some other zero-valued integer expression (hence the "implementation defined" language in the standard). The null pointer value may be something other than 0.

9. Since 0 is used to represent the null pointer, can it be thought of as an address with all zero bits?

No, it can have 1 bits also

10. What is array of pointer? How it is declared?

An array of pointers is an array that consists of variables of pointer type, which means that the variable is a pointer addressing to some other element. Suppose we create an array of pointer holding 5 integer pointers; then its declaration would look like:

```
int *ptr[5]; // array of 5 integer pointer.
```

11. Explain the relationship between array and pointer

An array is represented by a variable that is associated with the address of its first storage location. A pointer is also the address of a storage location with a defined type

12. Why is the addition of two pointer is impossible?

Because pointers represent locations in the memory. Each pointer is the address of a block in the memory. Subtracting two pointers will give you the size of memory between them. Addition or any other mathematical operation on values of pointers has no meaning.

13. Explain the comparison of two pointer?

If two pointers point to non-static data members of the same union object, they compare equal (after conversion to void* , if necessary). If two pointers point to elements of the same array or one beyond the end of the array, the pointer to the object with the higher subscript compares higher.

14. What is the difference between arr and &arr where arr is an array name , though both displays the base address of the array?

arr is an integer pointer (int*) which points the first element of the array. ... &arr is a pointer to an entire array. So, if we move &arr by 1 position it will point the next block of 5 elements. arr is a pointer to the first element of the array.So, if we move arr by 1 position it will point the second element.

15. When to use pointer to a function?

A pointer to a function points to the address of the executable code of the function. You can use pointers to call functions and to pass functions as arguments to other functions. You cannot perform pointer arithmetic on pointers to functions.

16. What are uses of dynamic memory allocation?

Dynamic memory allocation is a process that allows to allocate memory while our program is running, as opposed to telling the computer exactly how much we'll need (and for what) ahead of time.

17. What happened if malloc() is called?

You can call the malloc function at any time, and it will request a block of memory from the heap. The operating system will reserve a block of memory for your program, and you can use it in any way you like.

18. Difference between malloc() and calloc()

malloc() and calloc() functions are used for dynamic memory allocation in the C programming language. The main difference between the malloc() and calloc() is that calloc() always requires two arguments and malloc() requires only one.

19. How does one pointer point to another pointer?

A pointer to a pointer is a form of multiple indirection, or a chain of pointers. When we define a pointer to a pointer, the first pointer contains the address of the second pointer, which points to the location that contains the actual value

20. Describe the two different ways to specify the address of an array element?

By using & operator or pointers

21. Explain the terms: a) Dangling pointer b) function pointer c) Garbage collection d) Stack e) Static memory allocation

a) Dangling pointers and wild pointers in computer programming are pointers that do not point to a valid object of the appropriate type
b) A function pointer, also called a subroutine pointer or procedure pointer, is a pointer that points to a function.
c) Garbage collection is the systematic recovery of pooled computer storage that is being used by a program when that program no longer needs the storage. This frees the storage for use by other programs (or processes within a program).
d) A stack is a linear data structure, collection of items of the same type.
e) Static memory allocation is an allocation technique which allocates a fixed amount of memory during compile time and the operating system internally uses a data structure known as Stack to manage this.

Practice!!!

Write a program that reads in up to 10 strings or to EOF, whichever comes first. Have it offer the user a menu with five choices: print the original list of strings, print the strings in alphabetical order, print the strings in order of increasing length, print the strings in order of

the length of the first word in the string, and quit. Have the menu recycle until the user enters the quit request. The program, of course should actually perform the promised tasks.

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

```
#include <string.h>

#include <stdlib.h>


//pretty sure using file pointers doesn't mean that whole solution is
done with pointers concept

//also, these codes don't work properly for long sentences

void print_og(){

   int i=0;

   char buf[100];


   FILE *fp= fopen("strings.txt", "r");


   if (fp == NULL) {

   printf("Unable to open file\n");

   return;

   }




  while(!feof(fp) && i<10)//since either it should read 10 strings or
till EOF,

  //whichever comes first

  {


     fgets(buf, sizeof(buf), fp);
```

```
      printf("%s", buf);


      i++;
    }


    fclose(fp);


    printf("\n\n");
}


void alpha(){
     int i=0, j=0, line=0;
     char buf[50][100], temp[100];


     FILE *fp= fopen("strings.txt", "r");


     if (fp == NULL) {
     printf("Unable to open file\n");
     return;
     }




    while(!feof(fp) && i<10)
    {
      fgets(buf[i], sizeof(buf[i]), fp);
      i++;
      line++;//in case EOF is reached before reading 10 strings
    }

    for(i=0;i<line;i++)
    {
       for(j=i+1;j<line;j++)
       {
          if(strcmp(buf[i],buf[j])>0)//since actually ASCII codes are being
compared,
          //strings starting with small letters will come at end only
          {
             strcpy(temp,buf[i]);
             strcpy(buf[i],buf[j]);
             strcpy(buf[j],temp);
          }
       }
    }

    printf("\nAlphabetical list is:\n");
    for(i=0;i<line;i++)
    {
      printf("%s",buf[i]);//each string already has \n inside file itself, so
no need of \n
  }

  fclose(fp);

  printf("\n\n");
}

void str_len(){
   int i=0, j=0, line=0;
   char buf[50][100], temp[100];

   FILE *fp= fopen("strings.txt", "r");

   if (fp == NULL) {
   printf("Unable to open file\n");
   return;
   }



  while(!feof(fp) && i<10)
  {
    fgets(buf[i], sizeof(buf[i]), fp);
```

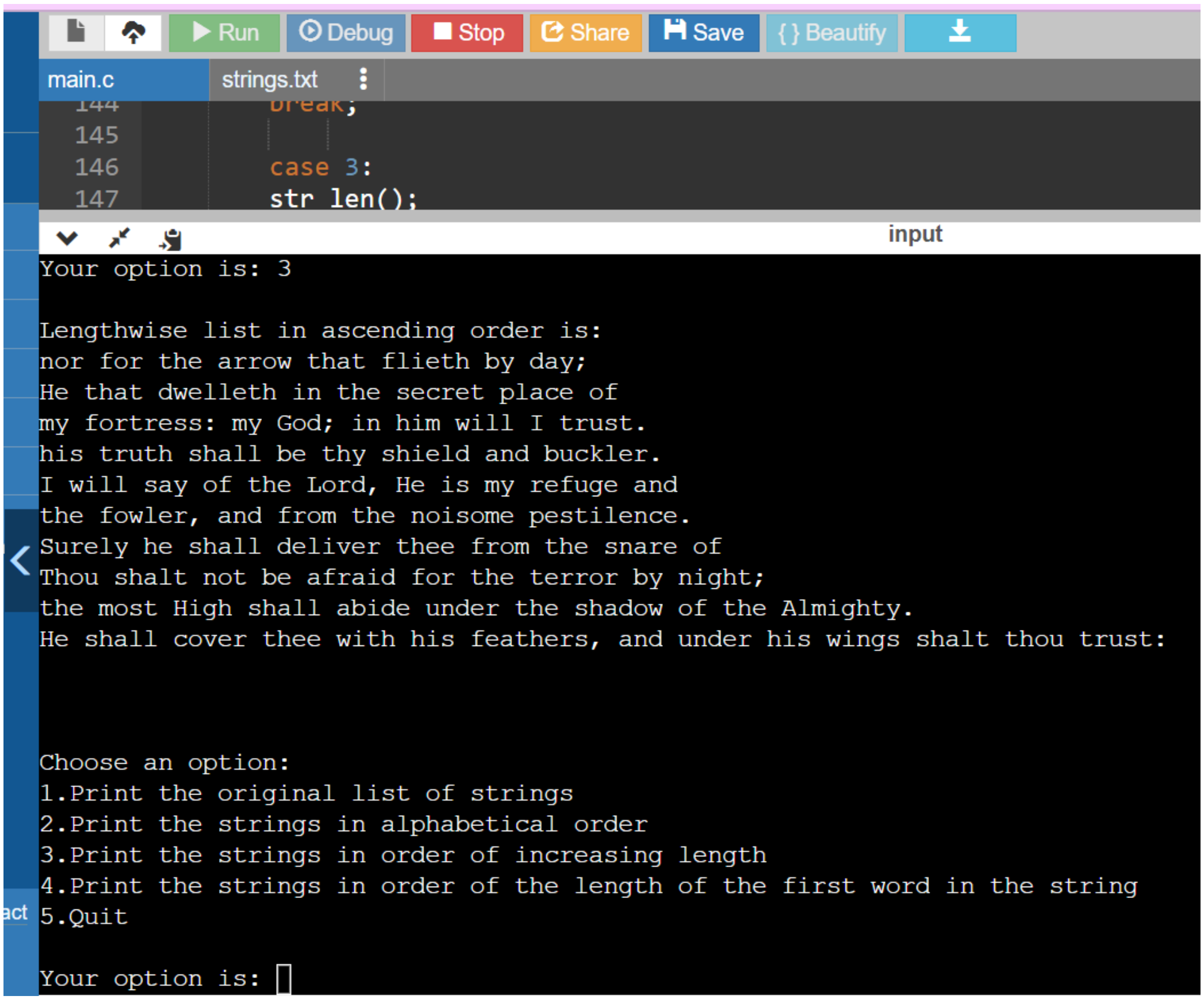
```
     i++;
     line++;//in case EOF is reached before reading 10 strings
   }

   for(i=0;i<line;i++)
   {
      for(j=i+1;j<line;j++)
      {
         if(strlen(buf[i])>strlen(buf[j]))
         {
            strcpy(temp,buf[i]);
            strcpy(buf[i],buf[j]);
            strcpy(buf[j],temp);
         }
      }
   }

   printf("\nLengthwise list in ascending order is:\n");
   for(i=0;i<line;i++)
   {
      printf("%s",buf[i]);//each string already has \n inside file itself, so
no need of \n
   }
```

```
  fclose(fp);


  printf("\n\n");
}


void word_len(){
   int i=0, j=0, line=0, count[50], k=0;
   char buf[50][100], temp[100];


   FILE *fp= fopen("strings.txt", "r");


   if (fp == NULL) {
   printf("Unable to open file\n");
   return;
   }



  while(!feof(fp) && i<10)
  {
    fgets(buf[i], sizeof(buf[i]), fp);
    i++;
    line++;//in case EOF is reached before reading 10 strings
  }
```

```
for(i=0;i<line;i++)
{
   for(j=0;j<strlen(buf[i]);j++)
   {
      if(buf[i][j]==' ')
      {
         count[i]=k;//length of first word of ith string
         k=0;//resetting for next string
         break;
      }

      else
        k++;
   }
}

for(i=0;i<line;i++)
{
   for(j=i+1;j<line;j++)
   {
      if(count[i]>count[j])
      {
          strcpy(temp,buf[i]);
          strcpy(buf[i],buf[j]);
```

```
            strcpy(buf[j],temp);
          }
      }
  }

  printf("\nFirst word lengthwise list in ascending order is:\n");
  for(i=0;i<line;i++)
  {
     printf("%s",buf[i]);//each string already has \n inside file itself, so
no need of \n
  }

  fclose(fp);

  printf("\n\n");
}

int main()
{
   int option;

do{
   printf("\nChoose an option:");
   printf("\n1.Print the original list of strings");
```

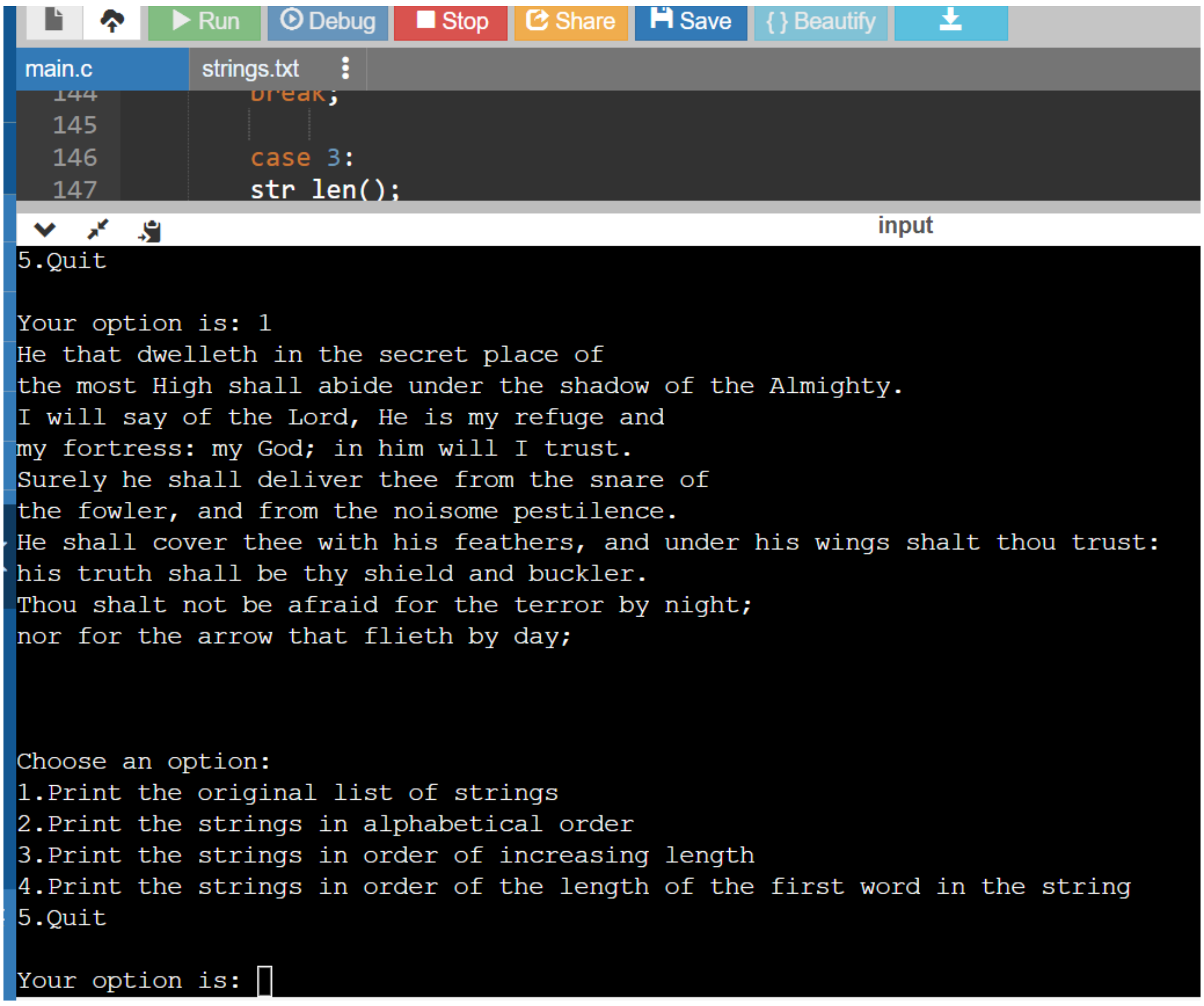
```
    printf("\n2.Print the strings in alphabetical order");
    printf("\n3.Print the strings in order of increasing length");
    printf("\n4.Print the strings in order of the length of the first word
in the string");
    printf("\n5.Quit");
    printf("\n\nYour option is: ");
    scanf("%d",&option);

    switch(option){
        case 1:
        print_og();
        break;

        case 2:
        alpha();
        break;

        case 3:
        str_len();
        break;

        case 4:
        word_len();
        break;
        case 5:
        exit(0);
        break;

        default:
        printf("\nChoose a correct option!");
        break;
    }

}while(1);



return 0;
}
```



```
         break;

         case 3:
         str_len();
```

```
Your option is: 3

Lengthwise list in ascending order is:
nor for the arrow that flieth by day;
He that dwelleth in the secret place of
my fortress: my God; in him will I trust.
his truth shall be thy shield and buckler.
I will say of the Lord, He is my refuge and
the fowler, and from the noisome pestilence.
Surely he shall deliver thee from the snare of
Thou shalt not be afraid for the terror by night;
the most High shall abide under the shadow of the Almighty.
He shall cover thee with his feathers, and under his wings shalt thou trust:



Choose an option:
1.Print the original list of strings
2.Print the strings in alphabetical order
3.Print the strings in order of increasing length
4.Print the strings in order of the length of the first word in the string
5.Quit

Your option is:
```

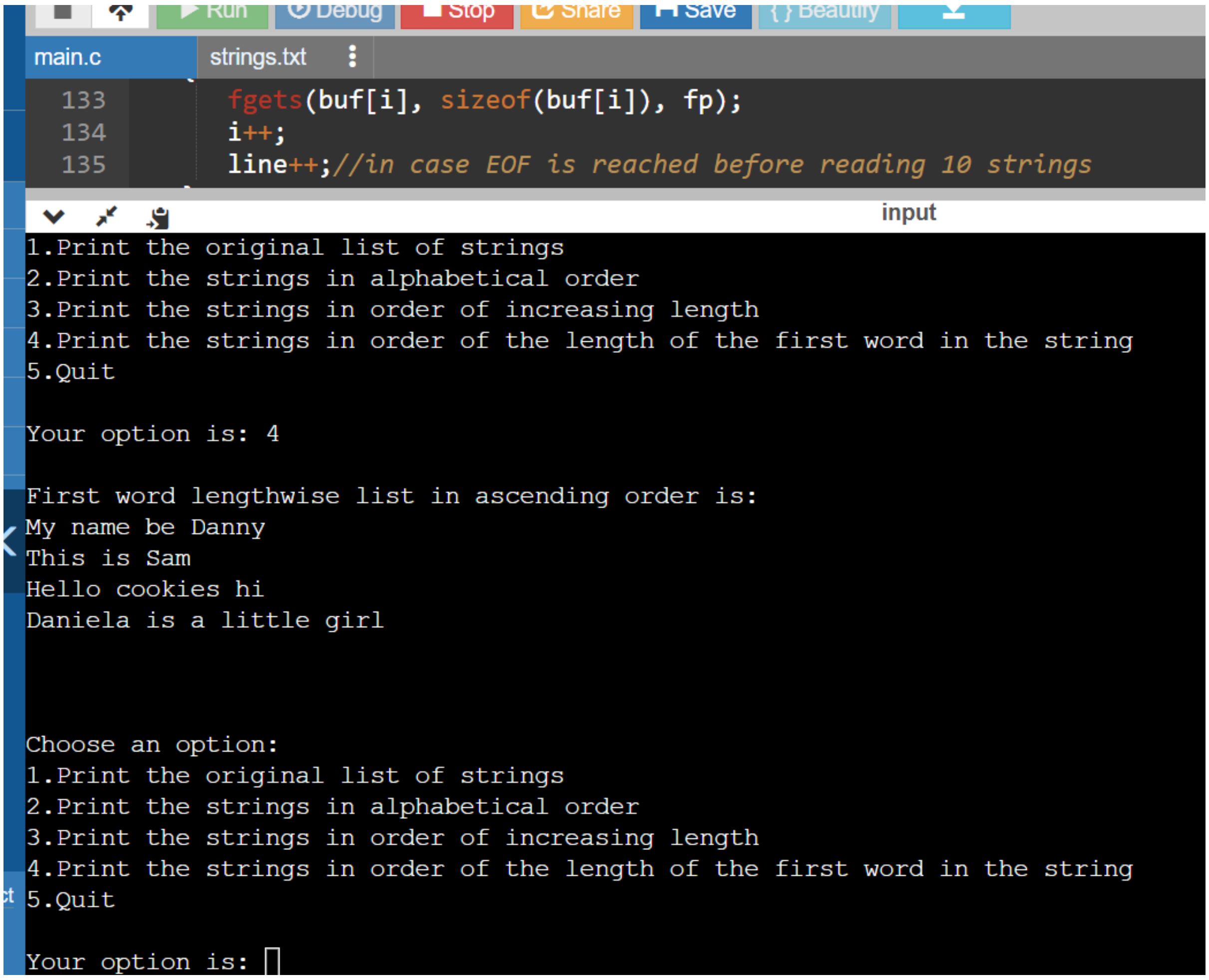
Samuela Abigail
71762108039



```
        break;

        case 3:
        str_len();
```

```
5.Quit

Your option is: 1
He that dwelleth in the secret place of
the most High shall abide under the shadow of the Almighty.
I will say of the Lord, He is my refuge and
my fortress: my God; in him will I trust.
Surely he shall deliver thee from the snare of
the fowler, and from the noisome pestilence.
He shall cover thee with his feathers, and under his wings shalt thou trust:
his truth shall be thy shield and buckler.
Thou shalt not be afraid for the terror by night;
nor for the arrow that flieth by day;



Choose an option:
1.Print the original list of strings
2.Print the strings in alphabetical order
3.Print the strings in order of increasing length
4.Print the strings in order of the length of the first word in the string
5.Quit

Your option is: 
```

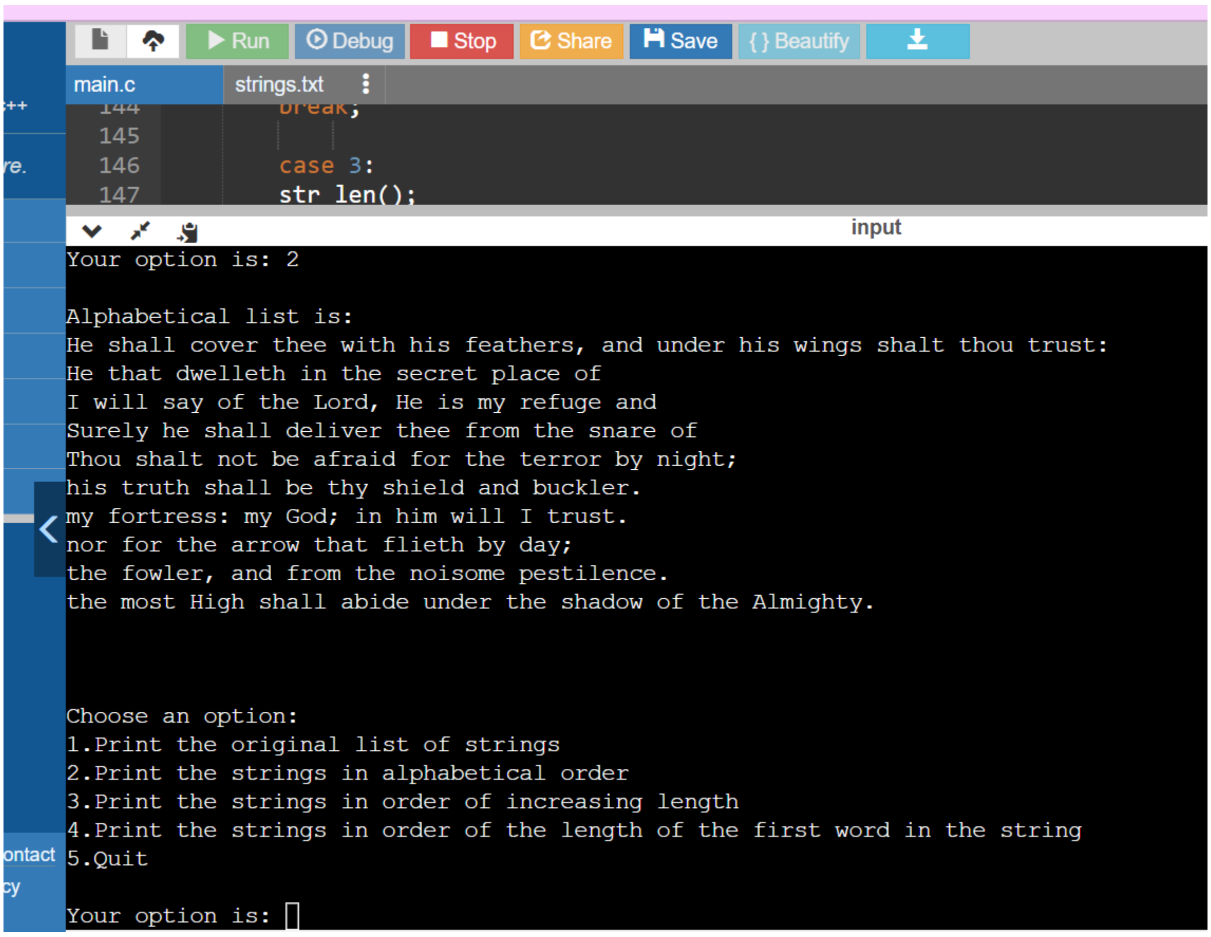
Samuela Abigail
71762108039



```c
fgets(buf[i], sizeof(buf[i]), fp);
i++;
line++;//in case EOF is reached before reading 10 strings
```

```
1.Print the original list of strings
2.Print the strings in alphabetical order
3.Print the strings in order of increasing length
4.Print the strings in order of the length of the first word in the string
5.Quit

Your option is: 4

First word lengthwise list in ascending order is:
My name be Danny
This is Sam
Hello cookies hi
Daniela is a little girl



Choose an option:
1.Print the original list of strings
2.Print the strings in alphabetical order
3.Print the strings in order of increasing length
4.Print the strings in order of the length of the first word in the string
5.Quit

Your option is: 
```

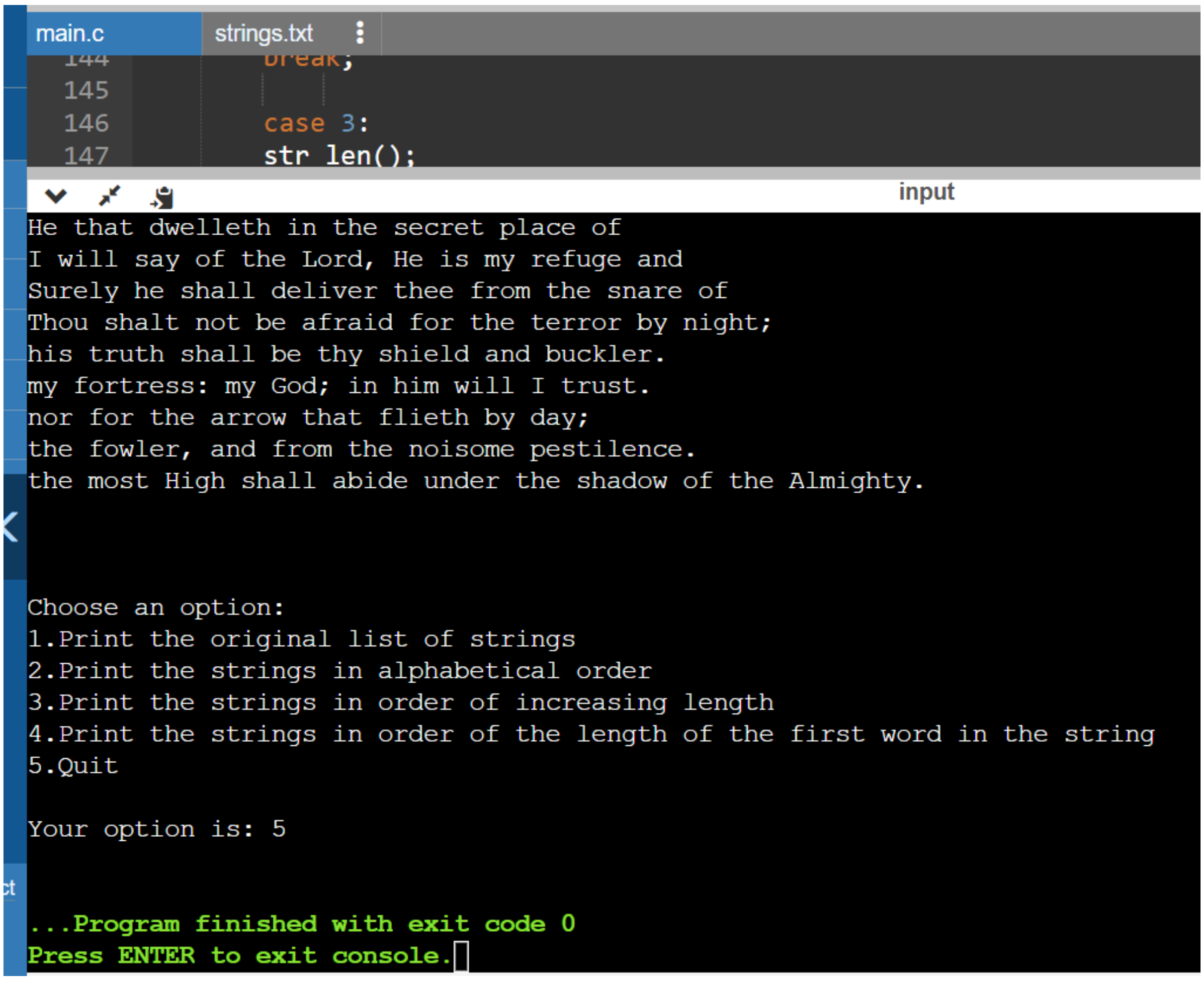
Samuela Abigail
71762108039



```
        break;

        case 3:
        str_len();
```

```
Your option is: 2

Alphabetical list is:
He shall cover thee with his feathers, and under his wings shalt thou trust:
He that dwelleth in the secret place of
I will say of the Lord, He is my refuge and
Surely he shall deliver thee from the snare of
Thou shalt not be afraid for the terror by night;
his truth shall be thy shield and buckler.
my fortress: my God; in him will I trust.
nor for the arrow that flieth by day;
the fowler, and from the noisome pestilence.
the most High shall abide under the shadow of the Almighty.



Choose an option:
1.Print the original list of strings
2.Print the strings in alphabetical order
3.Print the strings in order of increasing length
4.Print the strings in order of the length of the first word in the string
5.Quit

Your option is: 
```

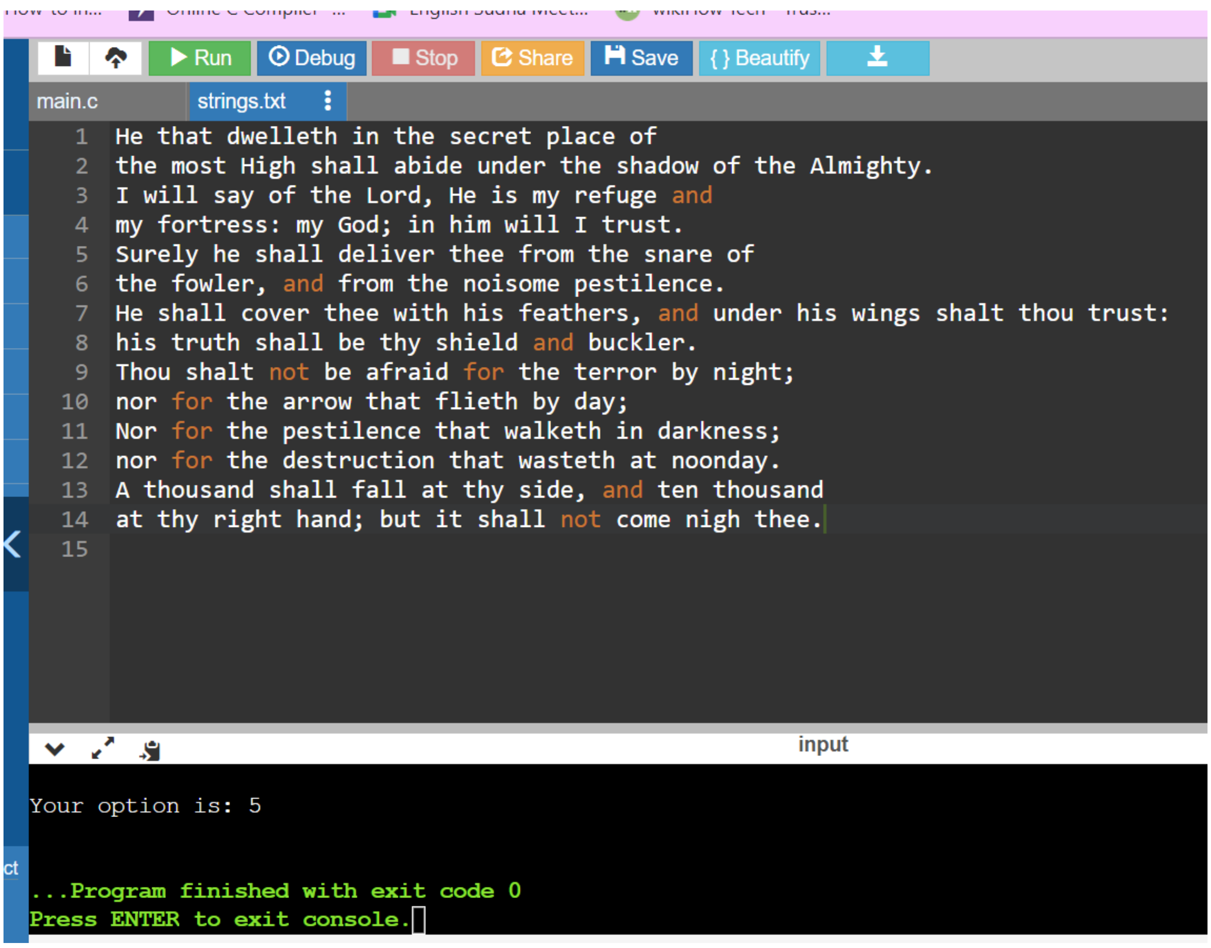
Samuela Abigail
71762108039



```
        break;

        case 3:
        str_len();
```

```
He that dwelleth in the secret place of
I will say of the Lord, He is my refuge and
Surely he shall deliver thee from the snare of
Thou shalt not be afraid for the terror by night;
his truth shall be thy shield and buckler.
my fortress: my God; in him will I trust.
nor for the arrow that flieth by day;
the fowler, and from the noisome pestilence.
the most High shall abide under the shadow of the Almighty.



Choose an option:
1.Print the original list of strings
2.Print the strings in alphabetical order
3.Print the strings in order of increasing length
4.Print the strings in order of the length of the first word in the string
5.Quit

Your option is: 5


...Program finished with exit code 0
Press ENTER to exit console.
```

Samuela Abigail
71762108039



```
He that dwelleth in the secret place of
the most High shall abide under the shadow of the Almighty.
I will say of the Lord, He is my refuge and
my fortress: my God; in him will I trust.
Surely he shall deliver thee from the snare of
the fowler, and from the noisome pestilence.
He shall cover thee with his feathers, and under his wings shalt thou trust:
his truth shall be thy shield and buckler.
Thou shalt not be afraid for the terror by night;
nor for the arrow that flieth by day;
Nor for the pestilence that walketh in darkness;
nor for the destruction that wasteth at noonday.
A thousand shall fall at thy side, and ten thousand
at thy right hand; but it shall not come nigh thee.
```

```
Your option is: 5


...Program finished with exit code 0
Press ENTER to exit console.
```

Samuela Abigail
71762108039