Date
Page

Special characters sets of C –

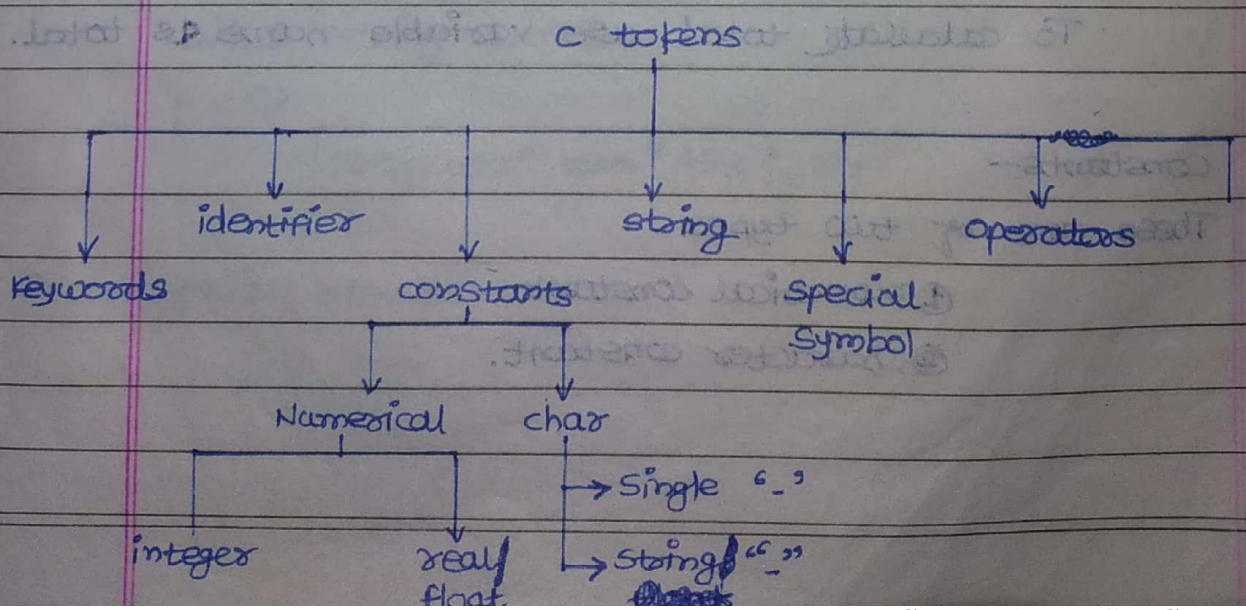
* The characters that can be used to form words, numbers & expressions depends upon the computer on which the program is run. However, a subset of characters is available that can be used on most personal micro, mini & mainframe computers.
* The character set are into following categories –
  1. Letters.
  2. Digits.
  3. Special characters. (,, &, ", %, etc).
  4. White spaces.

White space –

* These are non-printable characters they does not appears on screen. These are blank space, horizontal tab, carrage return, new line, etc.

C tokens –

* Tokens are nothing but individual word & different punctuation symbol used to write program.
* C has 6 types of tokens as bellows –



addoss
science-college ×
science_college ✓

① Keywords –

* These are words that have predefined meaning by the language.
* All keywords must be written in lower case. The different keywords are as bellows –

 if, else, for, while, int, float, switch, case, void, goto, etc.

② Identifiers –

* These are name of variable, array or function.
 These are user defined names and consists a sequence of character & digits.
* Both uppercase & lowercase are permitted but usually lowercase is used.

Rules for identifiers –

* First character must be alphabet or under score.
* Must contains only letters, digits or underscore.
* It should not be a keyword.
* It should not contain any white space (blank space).
* Usually, while writting program variables should be meaningful.

 eg.

 To calculate total use variable name as total.

③ Constants –

* These are of two types

 ① numerical constant.

 ② character constant.

① Numerical constant –

ⓐ Integer constant –

* It consists the number without decimal point.

 eg. 125, -125, etc.

ⓑ Real constant –

* It contains integer numbers with decimal point.
 These are also known as float.

 eg.

 95.23, 1.20345, 50.00, etc.

② Character constant –

* These are single character or group of characters.
 Single character constants are written in single quote.

 eg.

 'a', '2', etc.

④ String constant –

* A group of character enclosed in double quote are called string.

 eg.

 "computer", "459", etc.

⑤ Special symbols –

data: unsigned, signed +ve & -ve no.

* These are different punctuation symbol (",",;,!,?,:,etc). Each symbol will treate as token.

⑥ Operator-

* Operators are used to perform some operations like mathematical operation like comparison, etc.
* The operator uses symbol like (+,-,<,>,÷,*, etc. each symbol treated as token.

Data types in c-

* The variety of data types allow the programmer to select type appropriate to needs of applicat^n & machine
  ① Primary data type.
  ② Derived data type.
  ③ User defined data type.

① Primary data type-

* These are divided into four different category. Integer, char, float & double.
* Integer data type stores integer value whereas float & double stores floating or fractional value (no with decimal point).
* character data type is used to store single character value or string.
* The range & size (in byte) of these data type is as shown bellows-

int	1 byte	-32768 to 32767
char	1 byte	-128 to 127
float	4 bytes	3.4 E-38 to 3.4E+38
double	8 bytes	1.7E-308 to 1.7E+308

3.4E-38 = 3.4×10^{-38}

* These data types stores only limited values. These data types are further divided into subcategories like unsigned & signed. and short & long.
* The table below shows the data types with their subcategories, their range & size.

int or signed int	2 bytes	-32768 to 32767
unsigned int	2 bytes	0 to 65535
short int/ signed short int	1 byte	-128 to 127
short unsigned int	1 byte	0 to 255
signed long int	4 bytes	-2147483648 to 2147483647
unsigned long int	4 bytes	0 to 4294967295
float	4 bytes	3.4E-38 to 3.4E+38
double	8 bytes	1.7E-308 to 1.7E+308
long double	10 bytes	3.4E-4932 to 1.1E+4932
signed char	1 byte	-128 to 127
unsigned char	1 byte	0 to 255

Declaration of variables-

* In c language each variable must be declare before using it in program the declarat^n does two things-

① It tells the compiler the main name of variable.
② It specifies the type of data i.e stored in the variables.

Syntax-

data type $v_1, v_2, \ldots;$

Structure of c program-

Header section
<global variable declaration>
main()
{
local variable declaration
Body of program
}

* C program contains different sections such as header, global declaration, main function, local variable declarat^n & actual body of c program.
* Header section consists of different header files which are containing different library functions.
* It has form like #include<stdio.h> where stdio.h is a header file.
* Global variable declaration includes global variable declarat^n. Global variables are those whose scope is within all the function.
* Main is user defined function that contains actual program.
* Local variable declaration consists of declaration of local variables.
* Local variables are those scope is within the function only.
* These are declared as

int a, b, c;
float avg;

* A body of c program consists of different statement of c language using this statement we develop the logic of c program.

eg.

```
#include<stdio.h>
main()
{ int a,b,c;
a=20; b=30;
c=a+b;
printf("%d",c);
}
```

I/O statement in c-

* c provides two basic input output statements.

① scanf statement (Input).
② printf statement (Output).

① scanf statement-

* It is basic input statement in c. It is used to assaign a value to a variable.

scanf("control string", &variable);

* where control string specifies the type of data which will be entre, by user & assaign it to a variable.
* commonly used control strings are

control string	data type
%d	integer
%f	float
%c	character
%s	string

× scanf("%d", a);
× scanf("%d", &myname);

* Variable is a name of variable to which the inputed value will be assaigned.
* Variable is prefixes with & symbol. We can use multiple variables also seperated by ;.

eg.

```
scanf("%d", &a);
scanf("%s, %f", &a, &b);
scanf("%d, %d", &a, &b);
```

② Printf statement-

* Printf is an output statement it is use to print value of variable of any message on the screen its syntax is

```
Printf("control string", variable);
        or
printf("message");
        or
printf("message + control string", variable);
```

* Where variable is a name of variable whose value will be printed.
* Control string specifies the data type of variable. We can also print message on spring using printf statement. You can also make combination of message & value of variable within printf statement.

eg.

```
Printf("%d", sum);
Printf("Enter the first number");
Printf("sum of no. is %d", sum);
printf("%d", &sum);  ×
```

Q. Area of circle—

```
# include <stdio.h>
main()
{
  int a, r, pi=3.14;
  printf("Enter Radius");
  scanf("%d", &r);
  a=(pi*r*r);
  printf("Area of circle is %d", a);
}
```

Q. Read 2 no. & print their sum—

```
# include <stdio.h>
main()
{
  int a, b, c;
  printf("Enter a first no.");
  scanf("%d", &a);
  printf("Enter a second no.");
  scanf("%d", &b);
  c=a+b;
  printf("the sum is %d", c);
}
```

Q. Accept marks of 3 subjects & calculate total & avg.—

```
# include <stdio.h>
main()
{
```

```
int phy, che, math, t, avg;
Printf("Enter the Marks of Physics");
Scanf("%d", &phy);
Printf("Enter the Marks of Chemistry");
Scanf("%d", &che);
Printf("Enter the Marks of Math");
Scanf("%d", &math);
t=(phy+che+math);
avg=(t/3);
Printf("The total mark is %d", t);
Printf("The average is %d", avg);
}
```

2. Accept celcius temp. & convert into fahrenheit-

```
#include <stdio.h>
main()
{
float c, f;
Printf("Enter the temp. in celcius");
Scanf("%d", &c);
f=1.8*c+32;
Printf("temp. in fahrenheit is %f", f);
}
```