

C Language Exam – Answers

How can we determine the specific size of a particular data type?

We use the sizeof() operator, e.g., sizeof(int) returns the number of bytes used by int.

How do we put the header files of other modules or libraries into the source file of the program ?

By using #include for system headers or #include "header.h" for local ones.

How do we approach data fields of the composite data type (struct) in C?

By using the dot operator for variables (structVar.field) or the arrow operator for pointers (ptr->field).

Describe how the function int doit(int r) is called and how are the data passed to it?

It's called by doit(5); and data are passed by value — a copy of the argument is made.

How do you dynamically allocate memory for storing a sequence of 20 values of the data type int?

Using malloc: int *p = malloc(20 * sizeof(int));

How do you then increase this dynamic field to save ten more items?

Using realloc: p = realloc(p, 30 * sizeof(int));

What is the NULL identifier?

A macro representing a null pointer — it points to nothing.

How are text strings represented in C?

As arrays of characters terminated by '\0' (null character).

What represents the void type?

It indicates 'no type' or 'no value', used for generic pointers or functions returning nothing.

Explain the difference between the variable and the pointer to a variable.

A variable stores a value; a pointer stores the memory address of that variable.

What are the characters used in C for output control?

Examples: '\n' (newline), '\t' (tab), '\r' (carriage return), '\b' (backspace).

How is the process of running a program implemented in C language (under a full operating system)?

The compiled binary is loaded into memory, main() is called, execution proceeds sequentially until exit().

How to get a pointer to a variable defined as double d = 12.3;

Use &d; for example: double *ptr = &d;

How is the pointer and array [] different ?

An array is a fixed block of memory; a pointer can be reassigned to different memory addresses.

What is the difference between a pointer to a constant and a constant pointer? Write their definition.

Pointer to constant: `const int *p;` → cannot modify *p. Constant pointer: `int *const p;` → cannot change p itself.

Define a variable length array s of the size n, which is retrieved from the standard input.

```
int n; scanf("%d", &n); int s[n];
```

Describe briefly the struct type used in C.

A struct groups variables (possibly of different types) under one name.

What C compilers do you know?

Examples: GCC, Clang, MSVC, Turbo C, TinyCC.

Describe the process of creating an executable program from C source files.

Preprocessing → Compilation → Assembly → Linking → Executable.

Explain the difference between the variable and the pointer to a variable?

A variable holds data; a pointer holds an address to that data.

Does const in the variable definition guarantee that it cannot change?

Yes, the compiler prevents modification, though const can be bypassed by casting.

For what is the keyword 'do' used? Make a short example.

It starts a do-while loop that executes at least once: `do { ... } while(cond);`

For what is the keyword 'while' used? Make a short example.

It creates a loop that executes while the condition is true: `while(i < 10) i++;`

For what is the keyword 'for' used? Make a short example.

It is a compact loop construct: `for(i = 0; i < 10; i++) printf("%d", i);`

For what is the keyword 'if' used? Make a short example.

It performs conditional execution: `if(x > 0) printf("positive");`

For what is the keyword 'case' used? Make a short example.

It defines a branch in a switch statement: `switch(n){case 1: ...; break;}`

For what is the keyword 'switch' used? Make a short example.

Used for multi-way branching: `switch(op){case '+': sum(); break;}`

For what is the keyword 'break' used? Make a short example.

It exits loops or switch statements early: while(1){if(x) break;}

How do you create a pointer to a variable, such as int?

int a; int *p = &a;

What is the program return value that indicates a successful execution and why?

Return value 0 from main() signals success to the OS.

How do you dynamically allocate memory in C language?

Use malloc(), calloc(), or realloc().

How do you release dynamically allocated memory in C language?

Use free(pointer).

Define the diagonal (unit) matrix 3x3 as a 2D int array.

int A[3][3] = {{1,0,0},{0,1,0},{0,0,1}};

What is the difference between a thread and a process?

A process has its own memory space; threads share memory within one process.

What is a critical section?

A code region that must not be executed by more than one thread simultaneously.

Code Practice Answers

char a[] = "1,2,3"; printf("%ld\n",sizeof(a));

6 (five chars + '\0')

char a[] = {1,2,3}; printf("%ld\n",sizeof(a));

3

char a[] = {123}; printf("%ld\n",sizeof(a));

1

char a[] = {}; printf("%ld\n",sizeof(a));

0

char a[] = ""; printf("%ld\n",sizeof(a));

1 (just '\0')

char a = 123; printf("%ld\n",sizeof(a++));

1 (sizeof is evaluated at compile time)

char a = 123; printf("%ld\n",sizeof(++a));

1

```
char *a = {}; printf("%ld\n", sizeof(a));
```

8 on 64-bit systems (pointer size)

```
char a[] = {65}; printf("%d\n", a[0]);
```

65

```
char a[] = {65}; printf("%c\n", a[0]);
```

A

```
char a[] = "65"; printf("%d\n", a[0]);
```

54 ('6' in ASCII)

```
char a[] = "65"; printf("%c\n", a[0]);
```

6

```
char a[] = "65"; printf("%d\n", a[3]);
```

0 (beyond end, reads '\0')

```
char a[] = "65"; printf("%c\n", a[3]);
```

Non-printable (null char)

```
char a[] = {65}; printf("%d\n", a[0]++);
```

65 then increments to 66

```
char a[] = {65}; printf("%d\n", ++a[0]);
```

66

```
char a[] = {65}; printf("%d\n", a[0] = 64);
```

64

```
char a[] = {65}; printf("%d\n", a[0] == 64);
```

0 (false)

```
char a[] = {65}; printf("%d\n", a[0] != 64);
```

1 (true)

```
char a[] = "65"; printf("%c\n", a[0]++);
```

6 (then becomes '7')

```
char a[] = "65"; fork(); printf("%d", a[0]);
```

Both parent and child print 54 ('6')