Time left 0:09:18

Question 1

Not yet answered

Marked out of 1.00

What is `*(a + 8)` equivalent to?

Select one:

- `&a[8]` cross out
- It depends on how many bytes an int occupies on the machine. cross out
- `a[0] + 8` cross out
- None of the above. cross out
- `a[8]` cross out

Clear my choice

Question 2

Not yet answered

Marked out of 1.00

Suppose we create a 2D array of `int` using the following declaration:

```
int a[30][5];
```

What element is `*(a + 3)[5]`?

Select one:

- No element in `a` cross out
- The entire 9th row of a. cross out
- `a[8][0]` cross out
- `a[3][5]` cross out

Clear my choice

Question 3

Not yet answered

Marked out of 1.00

What is the output of the following code?

```
#include <stdio.h>
void fun(char *p) {
    *(p+2) = 'o';
    printf("%s", p);
}
int main() {
    char a[] = "souper";
    fun(a);
    return 0;
}
```

- a. o cross out
- b. Passing array when expecting pointer causes undefined behavior cross out
- c. souper cross out
- d. sooper cross out
- e. There is a syntax/compilation error cross out

Clear my choice

Question 4

Not yet answered

Marked out of 1.00

Suppose we have the following declarations:

```
int a;
int *p = &a;
int **q = &p;
```

Which of the following are valid ways to read an `int` from `stdin` into `a`? There may be multiple correct answers.

Select all that apply:

- [ ] `scanf("%d", *a);` cross out
- [ ] `scanf("%d", *p);` cross out
- [ ] `scanf("%d", a);` cross out
- [ ] `scanf("%d", q);` cross out
- [ ] `scanf("%d", **q);` cross out
- [x] `scanf("%d", p);` cross out
- [ ] `scanf("%d", &p);` cross out
- [x] `scanf("%d", *q);` cross out
- [ ] `scanf("%d", &q);` cross out
- [x] `scanf("%d", &a);` cross out

Question 5

Not yet answered

Marked out of 1.00

Suppose we have declared an array of `int` using `int a[10] = {0}`, and another array using `int b[5] = {0}`. What will the expression `&a[3] - &b[1]` return?

Select one:

- ○ 2 cross out
- ○ The expression will cause an error. cross out
- ◉ The expression will result in undefined behaviour. cross out
- ○ `2 * sizeof(int)` cross out
- ○ 0 cross out

Clear my choice

Question 6

Not yet answered

Marked out of 1.00

Suppose we have the following declarations:

```
int a;
int *p = &a;
```

Which of the following are valid ways to read an `int` from `stdin` into `a`? There may be multiple correct answers.

Select all that apply:

- [ ] `scanf("%d", *p);` cross out
- [x] `scanf("%d", &a);` cross out
- [ ] `scanf("%d", *a);` cross out
- [ ] `scanf("%d", &p);` cross out
- [x] `scanf("%d", p);` cross out
- [ ] `scanf("%d", a);` cross out

Question 7

Not yet answered

Marked out of 1.00

Suppose we create a 2D array of `int` using the following declaration:

```
int a[30][5];
```

What element is `*a[3]`?

Select one:

- ( ) No element in `a` cross out
- (•) `a[3][0]` cross out
- ( ) The entire 4th row. cross out
- ( ) `a[0][3]` cross out

Clear my choice

Question 8

Not yet answered

Marked out of 1.00

If `a` is an `int` variable and a pointer `p` points to `a`, then which of the follow expressions will give us `a`? There may be multiple correct answers.

Select all that apply:

- [x] `**&p` cross out
- [x] `*&a` cross out
- [ ] `&p` cross out
- [x] `*p` cross out
- [x] `*&*p` cross out
- [ ] `&*a` cross out
- [ ] `&*&p` cross out

Question 9

Not yet answered

Marked out of 1.00

Suppose, given some `int` variable `a`, we have the following pointer declaration:

```
int *p = &a;
```

Which of the following is the expression `&*&*&*p` equivalent to (yes we intend to put so many & and * operators)? There may be multiple correct answers.

Select all that apply:

- [ ] `&&p` cross out
- [ ] `&p` cross out
- [ ] `*p` cross out
- [x] `p` cross out
- [x] `&(*p)` cross out

Question 10

Not yet answered

Marked out of 1.00

Suppose we have declared an array of `int` using `int a[10] = {0}`. What will the expression `&a[100] - &a[30]` return?

Select one:

- ◉ 70 cross out
- ○ The expression will result in undefined behaviour. cross out
- ○ The expression will cause an error. cross out
- ○ 0 cross out
- ○ `70 * sizeof(int)` cross out

Clear my choice

Question 11

Not yet answered

Marked out of 1.00

Consider the following function fragment:

```
void fun(int arg1) {
  int a = arg1 + 5;
  int *p = &a;
  ...
```

Which one of the following `return` statements will provide the calling function with a usable pointer to `a`?

Select one:

- None of the above return statements will provide us with a usable pointer. cross out
- `return &a;` cross out
- `return p;` cross out
- `return &arg1;` cross out

Clear my choice

Question 12

Not yet answered

Marked out of 1.00

Suppose we have the following declarations:

```
int a[10] = {0};
int *p = a;
```

What will the expression `*++p = 10;` do?

Select one:

- The expression will result in an error. cross out
- It will increment the value at `a[0]` and then set it to 10. cross out
- It will set `a[1]` to 10 and move `p` to point to `a[1]` cross out
- It will set `a[0]` to 10, and move `p` to point to `a[1]` cross out

Clear my choice

Question 13

Not yet answered

Marked out of 1.00

Assuming it compiles successfully, what is the output of the following program?

```
int x = 5;

void change(int x) {
  x = 15;
}

int main() {
  int x = 10;
  printf("%d", x);
  return 0;
}
```

Select one:

- 5 cross out
- 15 cross out
- 10 cross out

Clear my choice

Question 14

Not yet answered

Marked out of 1.00

Suppose we have declared an array of `int` using `int a[10] = {0}`. What will the expression `&a[8] - &a[3]` return?

Select one:

- 0 cross out
- The expression will result in undefined behaviour. cross out
- The expression will cause an error. cross out
- 5 cross out
- `5 * sizeof(int)` cross out

Clear my choice

Question 15

Not yet answered

Marked out of 1.00

Suppose we have the following declarations:

```
int a[10] = {0};
int *p = a;
```

What will the expression `++*p = 10;` do?

Select one:

- The expression will result in an error. cross out
- It will increment the value at `a[0]` and then set it to 10. cross out
- It will set `a[1]` to 10 and move `p` to point to `a[1]` cross out
- It will set `a[0]` to 10, and move `p` to point to `a[1]` cross out

Clear my choice

Last saved at 19:22:30