

Question 1: 5 points

This one came from a bank. Here's an example:

We know the following:

The type of `i` is `(unsigned int *)`.

```
i is 0x2cab2a74
&i is 0x7239a470
*i is 0x5a7fb918
The four bytes at address 0x2cab2a78 are 0x542cc170
The four bytes at address 0x5a7fb918 are 0x64bd38ac
The four bytes at address 0x5a7fb91c are 0x2eb42318
The four bytes at address 0x7239a474 are 0x4ba96d14
```

What is the output of `printf("0x%x\n", i[1]);`

Question 2: 20 points

This question concerns the state of the stack and variables in the following program, when `ly()` returns.

```
int ly(int qp, int qy)
{
    int mn;

    qp++;
    mn = b() + b();
    return mn+qp+qy;
}

int main()
{
    int mo, th;

    mo = 3;
    th = 4;

    th = ly(mo, 287);
    return 0;
}
```

When `main()` starts, the frame pointer and the stack pointer are equal to: `0xffffcb8`.
For the first 8 parts of the question, please use the following answer key:

CS360 Midterm 2021

A - The address of the "pop #8" instruction in main()
B - The address of the "ret" instruction in ly()
C - The address of the "ret" instruction in main()
D - The frame pointer of ly()
E - The frame pointer of main()
F - The spilled value of r2 in ly()
G - The stack pointer of ly()
H - The stack pointer of main()
I - The variable mn in ly()
J - The variable mo in main().
K - The variable qp in ly()
L - The variable qy in ly()
M - The variable th in main().
N - Unknown

The Questions: To answer, simply answer one question per line. Start the line with the question number and a space and then the answer. Just use a letter for the first 8 questions. And again, the first 8 questions refer to when ly() returns.

Question 1. What is at address 0xffffc9c on the stack?
Question 2. What is at address 0xffffca0 on the stack?
Question 3. What is at address 0xffffca4 on the stack?
Question 4. What is at address 0xffffca8 on the stack?
Question 5. What is at address 0xffffcac on the stack?
Question 6. What is at address 0xffffcb0 on the stack?
Question 7. What is at address 0xffffcb4 on the stack?
Question 8. What is at address 0xffffcb8 on the stack?
Question 9. What is the first instruction of main()?
Question 10. When main() returns, what is the value of sp?

Question 3: 16 points

Write a program that prints the number of distinct files in the current directory. What the word "distinct" means is that if two files are "distinct," then they are not hard links to each other:

I don't care about include files, so don't bother with them. So, two things to remember while you answer: First, set the "paragraph" of your text box to "Preformatted", so that you have a nice fixed-width font to work with. Second, you can resize the answer box by dragging the lower-right corner. It will make your life easier to make the answer box larger.

Useful prototypes and structs:

CS360 Midterm 2021

```
DIR *opendir(const char *name);
struct dirent *readdir(DIR *d);
int closedir(DIR *d);
int stat(const char *name, struct stat *b);

struct stat {
    mode_t    st_mode;    /* File mode (see mknod(2)) */
    ino_t     st_ino;     /* Inode number - you may assume that this is an integer. */
    dev_t     st_dev;     /* ID of device containing */
                    /* a directory entry for this file */
    dev_t     st_rdev;    /* ID of device */
                    /* This entry is defined only for */
                    /* char special or block special files */
    nlink_t   st_nlink;   /* Number of links */
    uid_t     st_uid;     /* User ID of the file's owner */
    gid_t     st_gid;     /* Group ID of the file's group */
    off_t     st_size;    /* File size in bytes */
    time_t    st_atime;   /* Time of last access */
    time_t    st_mtime;   /* Time of last data modification */
    time_t    st_ctime;   /* Time of last file status change */
                    /* Times measured in seconds since */
    long      st_blksize; /* Preferred I/O block size */
    long      st_blocks;  /* Number of 512 byte blocks allocated*/
};

struct dirent {
    off_t      d_off;     /* offset of next disk dir entry */
    unsigned long d_fileno; /* file number of entry */
    unsigned short d_reclen; /* length of this record */
    char        *d_name;  /* name */
};
```

Question 4: 16 points

Your job is to write the procedure "nconcat", which has the following prototype:

```
char *nconcat(const char **str);
```

The argument "str" is an array of C-style strings, whose last element is NULL. Your job is to create and return a string composed of all of the strings in "str", in the order in which they appear in "str", each separated by a single space.

For example, suppose that "str" is as follows:

```
{ "fred", "daisy", "binky", NULL }
```

Then nconcat should return:

```
"fred daisy binky"
```

You must use **strcpy()/strcat()**, and your program must be efficient.

As always, I don't care about includes, so don't include them. Remember to set your entry box to "preformatted" and resize it so that it's easier to read.

Useful prototypes:

```
char *strcpy(char *dest, const char *src);
char *strcat(char *dest, const char *src);
int strlen(const char *s);
void *malloc(size_t sz);
```

CS360 Midterm 2021

Question 5: 10 points

You are at a job interview, and you are given the following question. Please answer it (just one paragraph):

"We maintain a directory of content for each of our clients. Each client's content is stored in large files (1 to 10 MB). We have reserved the first 16 bytes of each file to be a content tag. One of our operations is to read the content flags for all of the files in a directory. Please evaluate whether to use `fread()` or `read()` to read each content tag."

Question 6: 3 points

Below is a definition for a struct named `a194`. Please tell me what `sizeof(struct a194)` returns.

```
struct a194 {
    char c0;
    int i0;
    int i1;
    int i2;
    double d0;
};
```

Question 7: 10 points

The ASCII for the character 'A' in hex is `0x41`. Please tell me the output of the following program:

```
#include
#include
#include

int main()
{
    char buf[12];
    int *ip;
    int i;

    ip = (int *) buf;
    for (i = 0; i < 12; i++) buf[i] = 0;

    strcpy(buf+1, "ABEF");
    printf("1. 0x%x\n", ip[0]);
    printf("2. 0x%x\n", ip[1]);
    buf[0] = 'C';
    buf[5] = 'D';
    printf("3. %s\n", buf);
    return 0;
}
```

As always, please remember to set the 'Paragraph' setting to 'preformatted'.

The prototype for `strcpy()` is:

```
char *strcpy(char *dest, const char *src);
```