CS140 Midterm Exam - October 11, 2005

Jim Plank

Question 1

Behold the following C program:

```
void b(int *ip, int **ipp)
{
  printf("0x%x\n", &ip);

  printf("0x%x 0x%x\n", ip, ipp);

  *ipp += 2;
  **ipp = 1;
  *ip = 17;

  return;
}

main()
{
  int a[10];
  int *p;
  int i;

  for (i = 0; i < 10; i++) {
    a[i] = 15-i;
  }

  p = &(a[2]);
  printf("0x%x 0x%x\n", p, &p);

  b(p, &p);

  p++;
  *p = 3;

  printf("0x%x\n", a);
  printf("0x%x\n", &p);
  printf("0x%x\n", p);
  for (i = 0; i < 10; i++) {
    printf("%d ", a[i]);
  }
  printf("\n");
}
```

When we run this program, the first two lines of output are:

```
0xbffff024 0xbffff044
0xbfffeff8
```

What are the remaining lines of output?

Do not answer here -- use a separate sheet!!!!!

Question 2

Which of the following implementations of **dll_insert_b()** work correctly (there may be more than one)?

A.

```
void dll_insert_b(Dllist n, Jval v)
{
  Dllist newn;

  newn = (Dllist)
     malloc(sizeof(struct dllist));

  newn->val = v;

  newn->flink = n->flink;
  newn->blink = n;
  n->flink = newn;
  n->blink = newn;
}
```

B.

```
void dll_insert_b(Dllist n, Jval v)
{
  Dllist newn;

  newn = (Dllist)
     malloc(sizeof(struct dllist));

  newn->val = v;

  newn->flink = n;
  newn->blink = n->blink;
  n->blink = newn;
  n->blink->flink = newn;
}
```

C.

```
void dll_insert_b(Dllist n, Jval v)
{
  Dllist newn;

  newn = (Dllist)
     malloc(sizeof(struct dllist));

  newn->val = v;

  newn->flink = n;
  n->blink = newn;
  newn->blink = n->blink;
  n->blink->flink = newn;
}
```

D.

```
void dll_insert_b(Dllist n, Jval v)
{
  Dllist newn;

  newn = (Dllist)
     malloc(sizeof(struct dllist));

  newn->val = v;

  newn->blink = n->blink;
  newn->flink = newn->blink->flink;;
  n->blink->flink = newn;
  n->blink = newn;
}
```

E.

```
void dll_insert_b(Dllist n, Jval v)
{
  Dllist newn;

  newn = (Dllist)
     malloc(sizeof(struct dllist));

  newn->val = v;

  n->blink->flink = newn;
  newn->blink = n->blink;
  newn->flink = n;
  n->blink = newn;
}
```

F.

```
void dll_insert_b(Dllist n, Jval v)
{
  Dllist newn;

  newn = (Dllist)
     malloc(sizeof(struct dllist));

  newn->val = v;

  newn->flink = n;
  newn->blink = n->blink;
  newn->blink->flink = newn;
  newn->flink->blink = newn;
}
```

The following 8-line file is going to be standard input for the next three programs.

```
A goddess on
the mountain top
Is burning like a
silver flame
The summit of
beauty and love
And Venus
is her name
```

Question 3

What is the output of the following program when the above file is standard input?

```
main()
{
  IS is;
  Dllist d, tmp;
  int i;

  d = new_dllist();
  i = 0;
  is = new_inputstruct(NULL);

  while (get_line(is) >= 0) {
    if (i % 2 == 0) {
      dll_append(d, new_jval_s(strdup(is->fields[0])));
    } else {
      dll_prepend(d, new_jval_s(strdup(is->fields[is->NF-1])));
    }
    i++;
  }
  dll_traverse(tmp, d) {
    printf("%s\n", tmp->val.s);
  }
}
```

Question 4

What is the output of the following program when the above file is standard input?

```
main()
{
  Queue q;
  Stack s;
  Jval v;
  IS is;
  int i;

  q = new_queue();
  s = new_stack();
  i = 0;
  is = new_inputstruct(NULL);

  while (get_line(is) >= 0) {
    if (i % 2 == 0) {
      queue_enqueue(q, new_jval_s(strdup(is->fields[0])));
    } else {
      stack_push(s, new_jval_s(strdup(is->fields[0])));
    }
    i++;
  }
  while (!queue_empty(q)) {
    v = queue_dequeue(q);
    printf("%s\n", v.s);
  }
  while (!stack_empty(s)) {
    v = stack_pop(s);
    printf("%s\n", v.s);
  }
}
```

Question 5

What is the output of the following program when the above file is standard input?

```
main()
{
  int i;
  IS is;
  char *x, *y;

  i = 0;
  is = new_inputstruct(NULL);

  while (get_line(is) >= 0) {
    x = strchr(is->text1, 'o');
    if (x == NULL) x = is->text1;
    y = strchr(is->text1, 's');
    if (y == NULL) y = is->text1;
    printf("%d\n", y - x);
  }
}
```

Prototypes

```
typedef union {
    int i;
    long l;
    float f;
    double d;
    void *v;
    char *s;
    char c;
    unsigned char uc;
    short sh;
    unsigned short ush;
    unsigned int ui;
    int iarray[2];
    float farray[2];
    char carray[8];
    unsigned char ucarray[8];
  } Jval;

Jval new_jval_i(int);
Jval new_jval_l(long);
Jval new_jval_f(float);
Jval new_jval_d(double);
Jval new_jval_v(/* void */);
Jval new_jval_s(char *);
```

```
typedef struct dllist {
  struct dllist *flink;
  struct dllist *blink;
  Jval val;
} *Dllist;

Dllist new_dllist();
void free_dllist(Dllist);

void dll_append(Dllist, Jval);
void dll_prepend(Dllist, Jval);
void dll_insert_b(Dllist, Jval);
void dll_insert_a(Dllist, Jval);

void dll_delete_node(Dllist);
int dll_empty(Dllist);
```

```
#define MAXLEN 1001
#define MAXFIELDS 1000

typedef struct inputstruct {
  char *name;
  FILE *f;
  int line;
  char text1[MAXLEN];
  char text2[MAXLEN];
  int NF;
  char *fields[MAXFIELDS];
  int file;
} *IS;

IS new_inputstruct(char *name);
int get_line(/* IS */);
void jettison_inputstruct(/* IS */);
```

```
typedef void *Queue;

Queue new_queue();
void queue_enqueue(Queue q, Jval v);
Jval queue_dequeue(Queue q, Jval v);
int queue_empty(Queue q);
```

```
typedef void *Stack;

Stack new_stack();
void stack_push(Stack s, Jval v);
Jval stack_pop(Stack s, Jval v);
int stack_empty(Stack s);
```

---